Essays on cooperation in resource allocation and scheduling

Grundel, S.

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Essays on Cooperation in Resource Allocation and Scheduling

Essays on Cooperation in Resource Allocation and Scheduling

Proefschrift

ter verkrijging van de graad van doctor aan Tilburg University op gezag van de rector magnificus, prof. dr. E.H.L. Aarts, in het openbaar te verdedigen ten overstaan van een door het college voor promoties aangewezen commissie in de aula van de Universiteit op

vrijdag 26 juni 2015 om 10.15 uur

door

Soesja Grundel

geboren op 3 september 1983 te Amersfoort.

Promotiecommissie:

Promotores:	prof. dr. Peter Borm prof. dr. Herbert Hamers
Overige leden:	dr. Arantza Estévez Fernández prof. dr. Henk Norde prof. dr. Hans Peters

Het onderzoek in dit proefschrift is uitgevoerd onder toezicht van de Avans Hogeschool en CentER, de Graduate School van Tilburg School of Economics and Management (TiSEM), Tilburg University.

Essays on Cooperation in Resource Allocation and Scheduling

Als je samen gaat werken...
dan ga je samen werken.

Fien Sommen, January 2015

Preface

Six years ago, Avans offered me the opportunity to start a PhD project at Tilburg University to "investigate cooperative processes related to cost or reward allocation problems stemming from initiatives to establish joint projects". Finalizing this project, I would like to thank everyone for their support and contributions.

First, I would like to thank Liz Chermin, the person it all started with. Although Liz was not involved with the entire process it is certain it would have never started without her commitment and firm belief in me. Also the support of Peter ter Horst has been an important factor for starting this project. Thank you both.

I am grateful to my supervisors, Herbert Hamers and Peter Borm, for the support and guidance both gave me during these years. 'Goeiesmorgens' Herbert, with his illegible handwriting and in particular his verbal comments. I really appreciated his openness and honesty, as well as our constructive discussions as time went on. Peters detailed corrections of linguistic errors, but also the proposed reformulation of definitions were undoubtedly an essential condition for finishing this dissertation. I persisted in sloppiness, but both gentlemen were generous with their time and patience while being my mentors. Thank you both for guiding the process for content and personal aspects.

My thanks also go to the committee members, Arantza Estévez Fernández, Henk Norde, and Hans Peters. I feel privileged to have them on my committee and I would like to thank them for their extensive feedback during

my pre-defense and for taking the time to read all chapters. I would like to thank Barış Çiftçi for his contribution to Chapter 5 and Tsuyoshi Adachi for his valuable comments to Chapter 3.

Next, I would like to thank all colleagues and friends I met the last six years at the university. First, Heidi, Anja, and Lenie for their kind help whenever I dropped by their office with a question. And, of course, my room mate Korine, for our conversations, whishing her all the best for the future. Also a thank-you to all members of the 'lunchtafel', for their talks about football, Boer-zoekt-vrouw, or TiSEM gossip. I want to thank Ruud for providing the answers to my LaTeX questions and helping me with the layout of this dissertation. Finally, thanks for all the laughs with John, Edwin, Marieke, Elleke, Marloes, and the dancing queens, Salima and Mirjam. I loved to dance with those two pretty girls.

Although working both at the peaceful environment of the university and the chaotic environment of Avans has been very challenging, the diversity kept me going. As I was very fortunate to have had so many nice and knowledgeable people to work with at Avans, I have to limit myself to mentioning only a few. First, I remember Rob von Harenberg, raising his pedantic finger. He taught me a lot about all aspects of supervising graduation students. I also learned much from all members of the SB team, both old and new style, about their views on education. Special thanks go to the 'poepgroep' for their encouragements and always being there when I felt like chatting. Marc surrounded by his women, Marjolein organizing everything, Marlon making the suggestive remarks, Nicole motivating everyone and Marlieke overreacting. It is a pleasure working with this group of people, who have proven to free themselves from every difficult situation. Especially carpooling delivers surprisingly many new insights...

I would like to thank all members of the various boards of AMBM in recent years for the opportunities offered to me. Thanks to Peter, Bartje, René, Helmi, and Ilse.

All the encouragements by my professional environment would however have been in vain without the support of my family and friends. First of all JAV-members: Gilles, Hanneke, Merel, Lieke, Kjeld, Taco, Patrick, Annemarie, Daan, Pieter, Suzanne, Sofia, Rutgher, Daphne, Garrelt, Sylvia, and Phileine. Secondly, to my great friend Monique. I am grateful for our friendship and the fun we had studying Econometrics together. I would like to thank my 'parents-in-law' Ton and Joke for helping me out whenever they can. Their warmth and support has been very important to me. I thank my parents, Stephan and Trudi, for always supporting me and giving me the opportunity to develop myself into the person I am today. My thanks also go to my sister Nienke, my brother Max, and my 'brother-in-law' Tim for their understanding, encouragement, and support. My deepest gratitude goes out to Bram. He makes life beautiful. His love and support is most important to me, together with the happiness we share with Fien and Tijl. I very much look forward to the many beautiful moments that lie ahead.

Soesja Grundel
Nuenen, April 2015

Contents

1 Introduction **1**

1.1 Cooperation in Resource Allocation and Scheduling 1

1.2 Overview 9

2 Resource Allocation Problems with Linear Reward Functions **11**

2.1 Introduction 11

2.2 Resource Allocation Problems 14

2.3 Resource Allocation Games 18

2.4 Compromise Stability and the Nucleolus 25

3 Resource Allocation Problems with Concave Reward Functions **35**

3.1 Introduction 35

3.2 Resource Allocation Problems 37

3.3 Resource Allocation Games 45

3.4 Concluding Remarks 53

4 River Sharing Allocation **55**

4.1 Introduction 55

4.2 River Sharing Allocation Problems 58

4.3 River Sharing Allocation Games 75

5 Family Sequencing and Cooperation **89**

5.1 Introduction 89

5.2 Family Sequencing Situations 92

5.3 Family Sequencing Games 103

5.4 Ordered Family Sequencing Games 114

Bibliography **121**

Author index **129**

Chapter 1

Introduction

1.1 Cooperation in Resource Allocation and Scheduling

In today's economy with declining profit margins and growing interest in corporate social responsibility, cooperation has become increasingly important. By merging activities or capital, companies want to reduce costs or increase revenues. Such collaborations can be found, e.g., in purchasing situations (Groote Schaarsberg (2014)), logistics (Cruijssen (2006)), and resource pooling (Karsten (2013)).

In this dissertation, two topics are discussed. First, cooperation in resource allocation is considered. In resource allocation, resources are divided among collaborating agents who value the resources differently. Here assignments of resources are considered such that the total joint reward is maximized. Secondly, cooperation in a single-machine scheduling problem is discussed. For scheduling problems the reordering of jobs with respect to an initial processing order is considered, such that the cost savings are maximized. Clearly, both topics involve optimization aspects.

Next to optimization, this dissertation also focusses on the sharing of the resulting maximum total joint reward or maximum cost savings among the participating agents. Cooperative game theory addresses the subject of devising such allocations in an adequate and fair way. The most commonly used model for this, is the model of transferable utility (TU) games. In a TU-game each coalition S of players is associated with a certain worth $v(S)$, which in general, corresponds to the monetary benefits this coalition can obtain without help from players outside the coalition. By convention, $v(\emptyset) = 0$. The analysis of a TU-game focuses on how to allocate the joint reward of the grand coalition of all agents, where the values of coalitions of agents serve as a benchmark. Many solution concepts are proposed in the literature, each of them having a different interpretation of a "fair" allocation. Prominent solution concepts are the core (Gillies (1953)), the Shapley value (Shapley (1953)), the nucleolus (Schmeidler (1969)), and the compromise value (Tijs (1981)).

Both topics in this dissertation are related to Operations Research (OR) games. OR games are concerned with the interrelation between operations research and cooperative game theory. First, it considers the optimization problem of finding a joint optimal structure like a network or processing order. Once the optimal structure is determined, the extra gains from cooperation are allocated by applying concepts of cooperative game theory or by creating a context specific allocation rule. Such a rule can be based either on desirable properties in the specific context at hand or on a kind of decentralized mechanism that prescribes an allocation on the basis of the algorithmic process along which a jointly optimal structure is established. A survey of OR games is provided by Borm et al. (2001) and Curiel (2008). They distinguish between the topics of connection (Granot and Huberman (1982), Koster (1999)), routing (Potters et al. (1992), Bruno et al. (2011)), production (Owen (1975)), inventory (Fiestras-Janeiro et al. (2011)), and scheduling (Curiel et al. (2002)).

The following scheduling problem, which falls within the class of OR games, illustrates the natural interrelation between operations research and cooperative game theory.

Example 1.1.1. Consider a factory producing water pumps. Currently, three jobs are scheduled, one for a farmer, one for a gardener and one for a manufacturer. These jobs, denoted by f, g, and m, need to be scheduled on a single machine. The processing time of a pump is the time the machine takes to process the pump, and varies per client. Per client, the costs are assumed to be linearly dependent on the total waiting time. For example, the cost function $c_m(z) = 3z$ indicates that if the pump for the manufacturer is finished at time 3, his cost equals 9. Table 1.1 provides all data regarding to this scheduling problem.

Client	Processing time	Cost function
Farmer f	3	$c_f(z) = z$
Gardener g	1	$c_g(z) = z$
Manufacturer m	2	$c_m(z) = 3z$

Table 1.1: Jobs in Example 1.1.1

Initially, the jobs are scheduled in the order (f, g, m) with total joint costs of $c_f(3) + c_g(4) + c_m(6) = 25$. This order is displayed in Figure 1.1.

Assume the factory wants to determine a processing order of jobs that minimizes the total joint costs. It turns out that the corresponding optimal order equals (m, g, f) with total waiting costs of $c_m(2) + c_g(3) + c_f(6) = 15$. Hence, a costs saving of 10 is obtained with respect to the initial order. The optimal order is also displayed in Figure 1.1.

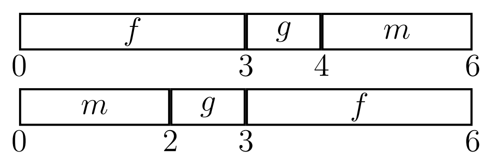

Figure 1.1: Initial order (f, g, m) and optimal order (m, g, f).

This optimization problem can be transformed into a multiple decision maker problem or game by supposing that each client (now also called a player) is charged for its costs. This problem gives rise to an OR game approach where on the one hand we need to determine the optimal processing order for the agents, and on the other hand, have to consider the allocation of cost savings to the players. In order to analyze this problem, Curiel et al. (1989) formally introduced corresponding sequencing TU-games. For sequencing games, the value of a coalition S of cooperating players is defined as the maximum cost savings the coalition S can achieve by means of reordering their jobs. It is assumed that a coalition S can only rearrange their jobs to an order for which for all jobs of players outside S the set of preceding jobs remains the same. In the context of the example, each player individually is not able to save costs and $\{f, m\}$, the coalition of the farmer and manufacturer, is not able to change the initial order due to the gardener's job scheduled in-between. Now, for example, coalition $\{f, g\}$ has two possible rearrangements: initial order (f, g, m), with total waiting costs $c_f(3) + c_g(4) = 7$, and (g, f, m) with total waiting costs $c_g(1) + c_f(4) = 5$. Hence, $\{f, g\}$ can save 2 by reordering their jobs. All coalitional values $v(S)$ are provided in the table below.

S	$\{f\}$	$\{g\}$	$\{m\}$	$\{f,g\}$	$\{f,m\}$	$\{g,m\}$	$\{f,g,m\}$
$v(S)$	0	0	0	2	0	1	10

The allocation of the total cost savings of 10 can, e.g., be done via the Equal Gain Splitting (EGS) rule, as introduced by Curiel et al. (1989), which is based on the fact that the optimal order can be obtained from the initial order by consecutive profitable switches of neighbors. According to the EGS rule each player is allocated half of the gains of all neighbour switches he is involved in, in order to reach an optimal order.

Here, the optimal order (m, g, f) can be reached in 3 neighbour switching steps from the initial order (f, g, m). First, reordering (f, g, m) to (g, f, m) involves the jobs of players f and g saving 2, which leads to allocation $(1, 1, 0)$. Next, reordering (g, f, m) to (g, m, f) involves the jobs of f and m saving 7, with allocation $(3\frac{1}{2}, 0, 3\frac{1}{2})$. Finally, the optimal order (m, g, f) is obtained from (g, m, f) by switching the jobs of g and m saving 1, with allocation $(0, \frac{1}{2}, \frac{1}{2})$. So, in total, the EGS rule leads to the allocation $(1, 1, 0) + (3\frac{1}{2}, 0, 3\frac{1}{2}) + (0, \frac{1}{2}, \frac{1}{2}) = (4\frac{1}{2}, 1\frac{1}{2}, 4)$. $\diamondsuit$

For the analysis of resource allocation problems, features from bankruptcy problems are used. Bankruptcy problems consider the allocation of a given amount of a perfectly divisible good (the estate) among a group of agents with rightful individual claims. Here the estate is insufficient to satisfy all the claims. For example, consider a firm going bankrupt, whose remaining assets do not cover the total demand of all creditors. The formal model of bankruptcy problems was first analyzed from a game-theoretic perspective by O'Neill (1982). The most common, rather pessimistic, translation of a bankruptcy problem into a cooperative game, is a bankruptcy game where the non-negative value of a coalition of cooperating agents equals the amount of the estate not claimed by agents outside the coalition. An overview on bankruptcy problems and associated games can be found in Thomson (2003) and is recently updated in Thomson (2015).

Example 1.1.2. Consider the following situation with a farmer, a gardener and a manufacturer each claiming water from a water reservoir nearby. Each agent requires a specific amount of water such that the sum of the claims exceeds the available amount of 4 liters of water. Assume that the farmer claims 3 liters of water, the gardener 1 and the manufacturer 2. Note that for this example the estate and claims are not monetary, but concerns a perfectly divisible good with the property "the more the better". The corresponding bankruptcy game v assigns to every coalition S of cooperating agents, a real number $v(S)$ which equals the amount of resources that is not claimed by the agents outside S. The values $v(S)$ for the various coalitions are given in the table below.

S	$\{f\}$	$\{g\}$	$\{m\}$	$\{f,g\}$	$\{f,m\}$	$\{g,m\}$	$\{f,g,m\}$
$v(S)$	1	0	0	2	3	1	4

In order to allocate the resources among the agents, several bankruptcy rules have been proposed in the literature. For example, the constrained equal award rule (CEA) allocates the estate as equal as possible among the agents, under the restriction that no agent should receive more than his claim. First, since the claim of the gardener is 1, each agent is assigned 1 liter of water. The remaining liter is divided equally among the farmer and manufacturer. Hence, the CEA rule leads to the allocation $(1,1,1) + (\frac{1}{2}, 0, \frac{1}{2}) = (1\frac{1}{2}, 1, 1\frac{1}{2})$. $\Diamond$

The agents involved in resource allocation problems, the subject of Chapter 2 and 3, optimize the assignment of resources and apply techniques from cooperative game theory in order to allocate the corresponding maximal total joint reward among the agents. In this setting, agents are characterized by an individual monetary reward function which allows for monetary compensations among agents, given a certain assignment of resources. Ideas from bankruptcy are used in the sense that the value of a particular coalition reflects the maximum total joint reward that can be derived from the amount of resources not claimed by agents outside the coalition. This approach is illustrated in the following example.

Example 1.1.3. Reconsider Example 1.1.2 where a farmer, a gardener and a manufacturer claim water from a water reservoir with size 4. Now each agent is characterized both by its claim and an individual (linear) reward function, which outlines the profit per liter of water. Moreover, it is assumed that agents cannot be assigned more than their claim. Table 1.2 provides all data regarding to the resource allocation problem. For example, the farmer claims 3 liters of water and has a monetary reward of 1 per liter of assigned water.

Agent	Claim	Reward function
Farmer f	3	$r_f(z) = z$
Gardener g	1	$r_g(z) = z$
Manufacturer m	2	$r_m(z) = 3z$

Table 1.2: Claims and Reward functions in Example 1.1.3.

It turns out that the total joint reward is maximized if the manufacturer obtains his claim of 2 liters and the remaining 2 liters are, e.g., assigned equally to the farmer and gardener. This yields a total joint reward of $r_f(1)+r_g(1)+r_m(2)=8$. To analyze the joint allocation problem of 8, a resource allocation game v is formulated which assigns to every coalition S a real number $v(S)$ which equals the maximum total joint reward for the coalition S using the amount of resources not claimed by the agents outside S. This is in line with the bankruptcy approach. The farmer on its own can use 1 liter of water since the gardener and manufacturer only claim 3 in total, with a reward of $r_f(1) = 1$; coalition $S = \{f, m\}$ can use 3 liters with a maximum total joint reward of $r_f(1)+r_m(2) = 7$. The coalitional values $v(S)$ are given in the table below.

S	$\{f\}$	$\{g\}$	$\{m\}$	$\{f,g\}$	$\{f,m\}$	$\{g,m\}$	$\{f,g,m\}$
$v(S)$	1	0	0	2	7	3	8

$\Diamond$

Papers using bankruptcy techniques to solve allocation problems in an OR setting are Estévez Fernández (2008) in the context of delays in projects, Read et al. (2014) in the allocation of (scarce) water resources, and Langar et al. (2015) in resource and power allocation in cooperative Femtocell networks.

River sharing allocation problems, the subject of Chapter 4, resemble resource allocation problems, but an additional structure is incorporated by means of a graph representing a river structure. Kilgour and Dinar (2001) and Ambec and Sprumont (2002) were the first to model this type of allocation problems as a cooperative TU-game. Moes (2013) provides an overview of river water allocation problems. Due to the structure of the river, there is a dependency relation among agents: the amount of water downstream agents can retrieve depends on the amount of water upstream agents retrieve. Similarly to OR games, optimization techniques are used for finding the jointly optimal assignment of water to the agents. For the allocation of the corresponding total joint reward, an adequate cooperative game is developed and analyzed.

Example 1.1.4. Reconsider the farmer, gardener and manufacturer of Example 1.1.3 with claims and reward functions as provided in Table 1.2. Now these agents are assumed to be located along a single-stream river from upstream to downstream. The inflow of the farmer, the most upstream agent, equals 3, and of the manufacturer, most downstream, equals 1. The gardener in between does not have any direct inflow. This river sharing allocation problem is displayed in Figure 1.2.

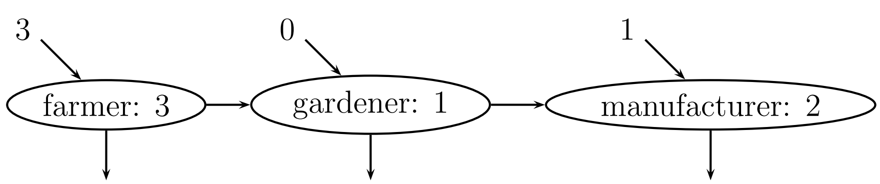

Figure 1.2: The river sharing allocation problem of Example 1.1.4

One can check that the total joint reward is maximized if the manufacturer is assigned 2 liters and the other 2 liters are assigned either to the farmer or gardener, as long as the assignment to the gardener does not exceed 1. The corresponding total joint reward equals 8. To allocate 8 among the agents a cooperative river sharing allocation game is analyzed in which the value $v(S)$ of a particular coalition S reflects the maximum total joint reward that can be derived from a cooperative assignment of water. Such an assignment may involve leakage to agents outside the coalition. Leakage occurs if upstream agents in coalition S transfer water to downstream agents in S with at least one agent in-between which is not in S. These intermediate agents are assumed to act individualistic by choosing their water use level in order to maximize individual payoffs by using all available water up to their claim. In particular, part of the transferred water from upstream agents in S will not arrive at the downstream agents in S. For such instances, agents in S have to consider whether the additional reward is sufficient to compensate the loss by leakage.

For coalition $S = \{f, m\}$, the farmer can guarantee to get 3 liters of water and the manufacturer can guarantee to get 1 liter of water with total joint reward $r_f(3) + r_m(1) = 6$. However, if the farmer transfers 2 liters of water to the manufacturer, the gardener in-between behaves individualistic and retrieves 1 liter, such that only 1 liter arrives to the manufacturer. Together with the manufacturer's own inflow, an assignment is obtained where 1 liter is assigned to farmer and 2 to the manufacturer with a total joint reward of $r_f(1) + r_m(2) = 7$. Hence, the loss of the gardener retrieving water is compensated by the extra marginal contribution of the manufacturer. The corresponding river sharing allocation game v is provided in the next table.

S	$\{f\}$	$\{g\}$	$\{m\}$	$\{f,g\}$	$\{f,m\}$	$\{g,m\}$	$\{f,g,m\}$
$v(S)$	3	0	3	3	7	3	8

$\diamondsuit$

Family sequencing, the subject of Chapter 5, extends the model of Curiel et al. (1989), as illustrated in Example 1.1.1, by considering a single machine scheduling problem for which setup times are involved. Jobs are processed according to an initial processing order and can be classified into distinct families with respect to their production requirements. A job does not require a setup when following another job from the same type, but a setup time is required when it follows a member of another type.

1.2 Overview

This dissertation is the collection of four self-contained essays, two of which are published. Chapter 2 is based on Grundel et al. (2013a), which is published in *Mathematical Methods of Operations Research*. Chapter 5 is based on Grundel et al. (2013), and is published in *European Journal of Operational Research*. Chapter 3 appeared as a discussion paper (Grundel et al. (2013b)). Because there is a certain overlap between the topics covered in the various essays, some concepts are defined multiple times over the various chapters. In the remainder of this section, we present an overview of this dissertation by means of the abstracts of the four essays. An extensive introduction of each of the different topics can be found at the beginning of each chapter.

Chapter 2 presents an extension of the traditional bankruptcy problem. In a resource allocation problem there is a common-pool resource, which needs to be divided among agents. Each agent is characterized by a claim on this pool and an individual linear monetary reward function for assigned resources. An assignment of resources is optimal if the total joint reward is maximized. Analyzing these problems a new class of transferable utility games is introduced, called resource allocation games. These games are based on the bankruptcy model, as introduced by O'Neill (1982). It is shown that the properties of totally balancedness and compromise stability can be extended to resource allocation games, although the property of convexity is not maintained in general. Moreover, an explicit expression for the nucleolus of these games is provided.

In Chapter 3 the model of Chapter 2 is generalized by characterizing agents with concave instead of linear reward functions. We provide a necessary and sufficient condition for optimality of an assignment. Analyzing the associated allocation problem of the maximal total joint reward, we consider corresponding resource allocation games. It is shown that these games have a non-empty core and thus allow for stable allocations. Moreover, an explicit expression for the nucleolus of these games is provided.

Chapter 4 analyzes river sharing allocation problems in which a set of agents is located along a river, sharing the available water. Each agent is characterized by a claim of water and a strictly increasing, differentiable, and concave reward function on the amount of retrieved water. An assignment of water is optimal if the total joint reward is maximized. We provide a necessary and sufficient condition for optimality of an assignment and provide a methodology to obtain such an assignment. Analyzing the associated allocation problem of the maximal total joint reward, we consider corresponding river sharing allocation games. A river sharing allocation game is defined where the value of a coalition reflects the maximal total joint reward a coalition can guarantee itself. The optimal assignment of coalitions may involve leakage, which occurs if intermediate individualists retrieve water up to their claim. It is shown that river sharing allocation games have a non-empty core.

Chapter 5 analyzes single-machine scheduling problems with family setup times both from an optimization and a cost allocation perspective. In a family sequencing situation jobs are processed on a single machine, there is an initial processing order on the jobs, and every job within a family has an identical cost function that depends linearly on its completion time. Moreover, a job does not require a setup when preceded by another job from the same family while a family specific setup time is required when a job follows a member of some other family. Explicitly taking into account admissibility restrictions due to the presence of the initial order, we show that for any subgroup of jobs there is an optimal order, such that all jobs of the same family are processed consecutively. To analyze the allocation problem of the maximal cost savings, we define and analyze a so-called corresponding cooperative family sequencing game. Using nonstandard techniques we prove that each family sequencing game has a non-empty core by showing that a particular marginal vector belongs to the core. Finally, we specifically analyze the case in which the initial order is family ordered.

Chapter 2

Resource Allocation Problems with Linear Reward Functions

2.1 Introduction

The formal model of bankruptcy problems was first analyzed from a game-theoretic perspective by O'Neill (1982). In a *bankruptcy problem* a certain amount, the estate, has to be divided among a group of claimants. Each claimant has a justified claim on the estate such that the sum of these claims exceeds the available estate. The example originally given by O'Neill is that of the division of an estate amongst several heirs when the estate cannot meet all the deceased's commitments. Another example is that of a firm going bankrupt, whose remaining assets do not cover the total demand of all creditors.

Many rules have been proposed to fairly allocate the estate in bankruptcy problems. Some of these rules are based on the associated cooperative *bankruptcy game* where the worth of a coalition is equal to what is left of the estate if all other claimants would receive their demands. Aumann and Maschler (1985) proposed and characterized a rule that coincides with the nucleolus of this bankruptcy game. For an overview on bankruptcy rules we refer to Thomson (2003) which is recently updated in Thomson (2015).

The bankruptcy model is a general framework for various kinds of allocation problems and is applied to many cases such as cost-sharing (Moulin (1991)), taxation (Young (1988)), and apportionment of indivisible good(s) problems (Young (1995)).

More generally, an allocation problem arises if a common-pool resource needs to be divided among a group of claimants. Young (1995) introduced a general framework with the concept of a "type" of a claimant: "the type of a claimant is a complete description of the claimant for purposes of the allocation, and determines the extent of a claimants entitlement to the good". A claimant is therefore characterized by a complete description of the claimant in several dimensions or attributes. Hereby, the traditional bankruptcy model deals with all allocation problems in which there is one perfectly divisible estate to be distributed among agents who can be characterized by a single (one-dimensional) claim on that estate. Kaminski (2000), Calleja et al. (2005), and Bergantiños et al. (2010) extend the traditional bankruptcy model by characterizing each agent by a vector of monetary claims. In this chapter we also extend the traditional bankruptcy problem to an allocation problem by characterizing the agents in two attributes: a justified claim on the resource and a linear reward function which describes the monetary reward for assigned resources.

In a *resource allocation* (RA) *problem* there is a limited supplied resource which is perfectly divisible. A specific resource assignment leads to some total joint reward obtained by the agents. The aim is to find a *fair* allocation of the maximum joint reward.

To illustrate the idea, consider the following economy with a manufacturer, a farmer and a gardener, each claiming water from the water reservoir nearby. Each business requires a specific amount of water such that the sum of the claims exceed the available amount of water. The profit per liter of water varies from business to business. Therefore, the corresponding allocation problem considers two elements. First the available water must be allotted to the businesses. The second element is to find an allocation of the total joint profit. In particular, profits may be redistributed among the businesses such that businesses are compensated who cede their water to others.

The framework we propose in this chapter, which is based on Grundel et al. (2013a) is in fact applicable to the general field of water resource management. Water resource management often involves a multitude of different agents with different interests who put their claims on a common-pool resource. The Tennessee Valley Authority case (Ransmeier (1942)) is one of the earliest cases that offered game theorists an opportunity to examine a practical problem of cost allocation in a water resources development project. Straffin and Heaney (1981) outlined some basic cooperative game-theoretic principles embedded in the analysis of this case and translated the main cost allocation methods into 'game theory language'. In Carraro et al. (2005), Parrachino et al. (2006), and Zara et al. (2006) an elaborated review is provided for game-theoretic water conflict resolution studies. For a recent overview of the literature about game theory and water resources we refer to Madani (2010).

A game-theoretic analysis of RA-problems also falls within the framework of operations research games. These games are concerned with the combinatorial optimization problem of finding a joint optimal structure like a network or processing order. Once the optimal structure is determined, game theoretical tools are applied subsequently to analyze the allocation of the corresponding joint rewards or costs. A survey of operations research games is provided by Borm et al. (2001).

Also for RA-problems we first analyze a joint optimization problem, the maximization of total joint reward via an optimal assignment of resources. Secondly, the maximum total joint reward is allocated to the agents. This is done by analyzing the associated cooperative *resource allocation* (RA) *game*. For this game the value of a particular coalition reflects the maximum total joint reward that can be derived from the amount of resources not claimed by agents outside the coalition.

Clearly, RA-games generalize bankruptcy games. For RA-games we study which properties of bankruptcy games are preserved. Bankruptcy games are totally balanced, convex and compromise stable. It is shown that the properties of totally balancedness and compromise stability can be extended to RA-games, although the property of convexity is not maintained in general. Further it is known that the nucleolus (Schmeidler (1969)) of bankruptcy games can be computed by the Aumann Maschler rule (Aumann and Maschler (1985)). Moreover, we show that the nucleolus of RA-games can also be computed by using this rule.

This chapter is organized as follows. In Section 2.2 the formal model of resource allocation problems is described and the optimal assignments of resources are characterized. In Section 2.3, we introduce and analyze corresponding resource allocation games. Section 2.4 is devoted to compromise stability and the computation of the nucleolus.

2.2 Resource Allocation Problems

This section formally introduces *resource allocation* (RA) *problems.* After introducing the model, optimal assignments of resources, in which the total joint reward is maximized, are characterized.

A *bankruptcy problem* is a triple (N, E, d), where N represents a finite set of claimants, $E \geq 0$ is the estate which has to be divided among the claimants, and $d \in \mathbb{R}^N_{++}$ is a vector of demands, where for $i \in N$, d_i represents agent i's claim on the estate. To justify the term 'bankruptcy' it is assumed that $\sum_{j \in N} d_j \geq E$.

An RA-problem extends a bankruptcy problem (N, E, d), with an estate E of some resource (e.g., water) and a demand vector d on this resource, by adding a *reward function* in the following way. For every agent $i \in N$ there exists a reward function $r_i : \mathbb{R}_+ \to \mathbb{R}_+$ describing the monetary gain to agent i from assigned resources. For every $z \in \mathbb{R}_+, r_i(z)$ denotes the monetary reward for agent i if he is assigned z units of resource. Now, an RA-problem can be summarized by (N, E, d, r). In this chapter the focus is on linear reward functions. Hence, for every $i \in N$ there exists an $\alpha_i \in \mathbb{R}_{++}$ such that $r_i(z) = \alpha_i z$. Hereby, for simplicity, RA-problems in this chapter will be referred to as (N, e, d, α). The class of all RA-problems with set of agents N is denoted by RA^N.

An outcome for an RA-problem consists of two elements: an *assignment* $x(N, e, d, \alpha)$ of resources and an *allocation* $y(N, e, d, \alpha)$ of the associated monetary reward. Throughout this chapter 'assignment' refers to the distribution of resources (e.g., water) and 'allocation' to the distribution of rewards (e.g., money). For the remainder of this chapter, $x(N, e, d, \alpha)$ is denoted by x and $y(N, e, d, \alpha)$ is denoted by y. Formally, a solution $f : RA^N \to \mathbb{R}^N \times \mathbb{R}^N$ is defined by

$$\begin{aligned}
f(N,e,d,\alpha) =&(x,y) && \\
&s.t. \sum_{j\in N} x_j = E && && (2.1)\\
&0 \le x_i \le d_i && \text{for all } i \in N && (2.2)\\
&\sum_{j\in N} y_j = \sum_{j\in N} r_j(x_j) && && (2.3)\\
&y_i \ge 0 && \text{for all } i \in N && (2.4)
\end{aligned}$$

for all $(N,e,d,\alpha) \in RA^N$. Constraint (2.1) tells us that the sum of assigned resources is equal to the estate. Constraint (2.2) ensures that the assigned resources do not exceed the individual demand and are non-negative for all agents in N. Let $F(N,e,d,\alpha)$ be the set of *feasible* assignments x determined by conditions (2.1) and (2.2), i.e.

$$F(N,e,d,\alpha) = \left\{ x \in \mathbb{R}^N \middle| \sum_{j\in N} x_j = E, 0 \le x_i \le d_i \text{ for all } i \in N \right\}.$$

Constraint (2.3) ensures that the total reallocated sum of rewards is equal to the total joint reward obtained from the underlying assignment x. Assume that no costs are involved for transferring rewards. One specific type of solution is the *direct solution*. A solution f is called direct, if for all $(N,e,d,\alpha) \in RA^N$, we have that $f(N,e,d,\alpha) = (x, r(x))$ for some $x \in F(N,e,d,\alpha)$, where $r(x) = (\alpha_i x_i)_{i\in N}$ is the direct reward vector with respect to x.

Example 2.2.1. Consider an RA-problem $(N,e,d,\alpha) \in RA^N$ with $N = \{1,2,3\}$ and estate $E = 4$. Assume that agent 1 claims 3 units of resource, agent 2 claims 1 unit, and agent 3 claims 2. Therefore, the demand vector is $d = (3,1,2)$. Note that the sum of these demands exceeds the available estate such that not all agents can obtain their full demand. With reward vector $\alpha = (1,1,3)$, e.g., the reward function of agent 1 is given by $r_1(z) = z$ for z such that $0 \le z \le d_1$. The feasible assignment $x = (2,1,1)$ has total joint reward 6 and $r(x) = (2,1,3)$. Note that $r(x)$ but also e.g., $y = (2,2,2)$, satisfy conditions (2.3) and (2.4). $\Diamond$

The set of feasible assignments is large and there are many possible (re)allocations of the corresponding rewards. Throughout this article, assignments of resources which maximize the total joint reward are considered. The remainder of this section is dedicated to finding these *optimal assignments.*

Let $(N, e, d, \alpha) \in RA^N$. The maximum total joint reward $v(N, e, d, \alpha)$ is determined by

$$v(N, e, d, \alpha) = \max \left\{ \sum_{j \in N} r_j(x_j) \middle| x \in F(N, e, d, \alpha) \right\}.$$

The set $X(N, e, d, \alpha)$ of optimal assignments is given by

$$X(N, e, d, \alpha) = \left\{ x \in F(N, e, d, \alpha) \middle| \sum_{j \in N} r_j(x_j) = v(N, e, d, \alpha) \right\}.$$

Before characterizing the set $X(N, e, d, \alpha)$ of optimal assignments we introduce some additional notation. Write $\{\alpha_j | j \in N\} = \{\beta_1, \beta_2, \ldots, \beta_m\}$ with $\beta_1 > \beta_2 > \cdots > \beta_m$ and define

$$N_l = \{j \in N | \alpha_j = \beta_l\}$$

for all $l \in \{1, \ldots, m\}$. Hence, each set of agents N_l contains all agents $i \in N$ with reward parameter $\beta_l = \alpha_i$. Denote the total demand of agents in N_l as $d(N_l)$, i.e. $d(N_l) = \sum_{j \in N_l} d_j$. Thus, this new notation is used to 'aggregate' agents with equal reward parameters and sort those in decreasing order by β. Define the *pivot index* $k \in \{1, \ldots, m\}$ such that

$$\sum_{l=1}^{k-1} d(N_l) \leq E < \sum_{l=1}^{k} d(N_l). \tag{2.5}$$

If $E = \sum_{j \in N} d_j$, then we assume $k = m$. With pivot index k define three disjoint sets P_1, P_2 and P_3 of agents in the following way

$$\begin{aligned} P_1 &= \bigcup_{l=1}^{k-1} N_l; \\ P_2 &= N_k; \\ P_3 &= \bigcup_{l=k+1}^{m} N_l. \end{aligned}$$

Note that P_1 and P_3 may be empty, but P_2 is always non-empty. Hence, P_1 contains all agents with reward parameters greater than β_k, and P_3 all agents with reward parameters lower than β_k.

Example 2.2.2. Reconsider the RA-problem of Example 2.2.1. Clearly, there are two different reward parameters, i.e. $m = 2$, $N_1 = \{3\}$ with $\beta_1 = \alpha_3 = 3$ and $N_2 = \{1, 2\}$ with $\beta_2 = \alpha_1 = \alpha_2 = 1$. Observe that

$$d(N_1) = 2 < E = 4 < d(N_1) + d(N_2) = 6.$$

Hence, the pivot index is $k = 2$. Consequently, $P_1 = N_1 = \{3\}$, $P_2 = N_2 = \{1, 2\}$, and $P_3 = \emptyset$. $\diamondsuit$

It is readily verified that the total joint reward is maximized, if the resources are assigned to agents in decreasing order of their reward parameter. By definition it holds that the estate suffices to meet the demand of all agents in P_1 and some of the demand of agents in P_2. Moreover, it holds that the reward parameters of agents in P_1 are greater then those in P_2, which have a greater reward parameter then agents in P_3. Consequently, assignments such that agents in P_1 are assigned their demand and agents in P_3 are assigned no resources, are optimal. These optimal assignments, are characterized in the following theorem.

Theorem 2.2.1. *Let $(N, e, d, \alpha) \in RA^N$ and $x \in F(N, e, d, \alpha)$. Then $x \in X(N, e, d, \alpha)$ if and only if $x_i = d_i$ for all $i \in P_1$ and $x_i = 0$ for all $i \in P_3$.*

Proof. Let $x \in X(N, e, d, \alpha)$. First suppose there exists an agent $i \in P_1$ such that $x_i < d_i$. This implies that

$$\sum_{j \in P_2 \cup P_3} x_j = E - \sum_{j \in P_1} x_j > E - \sum_{j \in P_1} d_j = E - \sum_{l=1}^{k-1} d(N_l) \geq 0.$$

We may conclude that there is at least one agent $j \in P_2 \cup P_3$ for which $x_j > 0$. Since $\alpha_j < \alpha_i$, the total joint reward would strictly increase if agent j transfers $\min\{d_i - x_i, x_j\}$ to agent i. This establishes a contradiction.

Secondly, suppose there exists an agent $i \in P_3$ such that $x_i > 0$. This implies that

$$E - \sum_{j \in P_1 \cup P_2} x_j = \sum_{j \in P_3} x_j > 0 > E - \sum_{l=1}^{k} d(N_l) = E - \sum_{j \in P_1 \cup P_2} d_j.$$

The first equality follows from (2.1), the first inequality from (2.2). The second inequality follows from (2.5), the last equality from the definitions of P_1 and P_2. This shows that $\sum_{j \in P_1 \cup P_2} x_j < \sum_{j \in P_1 \cup P_2} d_j$. Consequently, there is at least one agent $j \in P_1 \cup P_2$ whose demand is not fully satisfied, i.e. $x_j < d_j$. Since $\alpha_j > \alpha_i$, the total joint reward would increase if agent i transfers $\min\{d_j - x_j, x_i\}$ to agent j. This establishes a contradiction and proves the *only if* part.

All assignments $x \in F(N, e, d, \alpha)$ such that $x_i = d_i$ for all $i \in P_1$ and $x_i = 0$ for all $i \in P_3$ lead to the same total joint reward. Clearly, such assignment exists by the fact that $\sum_{j \in N} r_j(x_j)$ is a real-valued continuous function on a non-empty, compact interval. This finishes the proof. $\square$

Hence, the pivot index k is used to specify the set of agents N_k, the set which may not obtain the full demand in the optimal assignment of resources; all agents with larger reward parameters obtain their full demand, agents with a lower reward parameter obtain nothing.

Example 2.2.3. The set of optimal assignments of the RA-problem of Example 2.2.1 can be written as follows:

$$Conv\{(2, 0, 2), (1, 1, 2)\}.$$

$\diamondsuit$

As can be seen in Example 2.2.3, the optimal assignment is not necessarily unique. The RA-problems for which there is exactly one optimal assignment are characterized in the following corollary.

Corollary 2.2.2. *Let* $(N, e, d, \alpha) \in RA^N$. *Then* $|X(N, e, d, \alpha)| = 1$ *if and only if* $E = \sum_{j \in P_1} d_j$ *or* $|P_2| = 1$.

2.3 Resource Allocation Games

In this section we introduce the class of *resource allocation* (RA) *games.* A *transferable utility* (TU) *game* is an ordered pair (N, v) where N is the finite set of agents, and v the characteristic function on 2^N, the set of all subsets of N. The function v assigns to every coalition $S \in 2^N$ a real number $v(S)$

with $v(\emptyset) = 0$. Here, $v(S)$ is called the worth or value of the coalition S. The set of all TU-games with set of agents N is denoted by TU^N. Where no confusion arises, we write v rather than (N, v).

Consider a bankruptcy problem (N, E, d). For the associated bankruptcy (BR) game $v_{E,d}$ the value of a coalition S is determined by the amount of E that is not claimed by agents in $N \setminus S$. Hence,

$$v_{E,d}(S) = \max\left\{0, E - \sum_{j \in N \setminus S} d_j\right\}$$

for all $S \in 2^N$.

Now consider an RA-problem (N, e, d, α). We assume that a coalition S can only use the amount of resources $D(S)$ not demanded by the agents in $N \setminus S$. Let $(S, D(S), d|_S, \alpha|_S) \in RA^S$ describe the associated resource allocation problem for S[1], where

$$D(S) = \max\left\{0, E - \sum_{j \in N \setminus S} d_j\right\},$$

Note that $D(N) = E$. By the fact that $D(S) \leq \sum_{j \in S} d_j$ it follows that $(S, D(S), d|_S, \alpha|_S)$ again is an RA-problem.

Corollary 2.3.1. *Let $(N, E, d, \alpha) \in RA^N$ and let $S \subset N$. Then $(S, D(S), d|_S, \alpha|_S) \in RA^S$.*

In the RA-game v^R, associated to an RA-problem (N, e, d, α), the worth of coalition S equals

$$v^R(S) = v\left(S, D(S), d|_S, \alpha|_S\right).$$

The class of RA-games extends the class of BR-games, i.e. every BR-game can be written as an RA-game in which $\alpha_i = \alpha_j$ for all $i, j \in N$.

Let $S_l = S \cap N_l$ for all $l \in \{1, \ldots, m\}$. We extend the definition of the pivot index, as provided in (2.5) for the grand coalition, to every possible

[1] $d|_S \in \mathbb{R}^S$ denotes the restricted vector of demands (reward parameter) for agents in S with respect to $d \in \mathbb{R}^N$, i.e. $d|_s = (d_i)_{i \in S}$.

coalition S as follows. The *pivot index* $k(S)$ is such that

$$\sum_{l=1}^{k(S)-1} d(S_l) \leq D(S) < \sum_{l=1}^{k(S)} d(S_l). \tag{2.6}$$

If $D(S) = 0$, then we assume that $k(S) = 1$ and if $D(S) = \sum_{j \in S} d_j$, then we assume $k(S) = m$.

Let $x^S \in X(S, D(S), d|_S, \alpha|_S)$ be an optimal assignment of resources to agents in S. Theorem 2.2.1 tells us that $x_i^S = d_i$ for all $i \in S_1 \cup \cdots \cup S_{k(S)-1}$ and $x_i^S = 0$ for all $i \in S_{k(S)+1} \cup \cdots \cup S_m$. The remaining resources $(D(S) - \sum_{l=1}^{k(S)-1} d(S_l))$ are assigned to agents in $S_{k(S)}$. Clearly, $v^R(S)$ equals the total direct reward of agents in S associated to x^S. This allows us to construct an explicit formula for $v^R(S)$.

Theorem 2.3.2. *Let $(N, e, d, \alpha) \in RA^N$ and let v^R be the associated RA-game. Then*

$$v^R(S) = \sum_{l=1}^{k(S)-1} \beta_l d(S_l) + \beta_{k(S)} \left(D(S) - \sum_{l=1}^{k(S)-1} d(S_l) \right)$$

for all $S \in 2^N \setminus \{\emptyset\}$.

Example 2.3.1. Reconsider the RA-problem of Example 2.2.1 where $E = 4, d = (3, 1, 2)$ and $\alpha = (1, 1, 3)$. The corresponding values of $D(S)$, $k(S)$, and $v^R(S)$ are given in the table below.

S	1	2	3	1,2	1,3	2,3	N
$D(S)$	1	0	0	2	3	1	4
$k(S)$	2	1	1	2	2	1	2
$v^R(S)$	1	0	0	2	7	3	8

Firstly, we illustrate the underlying computations for coalition $S = \{1, 2\}$. Recall $m = 2$, $N_1 = \{3\}$ and $N_2 = \{1, 2\}$. Therefore, $S_1 = \emptyset$ and $S_2 = \{1, 2\}$. The amount of available resources equals

$$D(S) = \max\{0, 4 - 2\} = 2.$$

This yields,

$$\begin{array}{ccc} d(S_1) & < D(S) < & d(S_1) + d(S_2) \\ 0 & < 2 < & 0 + 4 \end{array}$$

and consequently, $k(S) = 2$. Therefore,

$$\begin{aligned} v^R(12) &= \beta_1 d(S_1) + \beta_2(D(S) - d(S_1)) \\ &= 3 \cdot 0 + 1(2 - 0) = 2. \end{aligned}$$

Secondly, consider $S = \{2\}$. Then $D(S) = \max\{0, 4 - 3 - 2\} = 0$. So, $k(S) = 1$ and $v^R(S) = 0$. $\Diamond$

From Example 2.3.1 we immediately see that RA-games are not *convex* in general since the condition

$$v(S \cup \{i\}) - v(S) \leq v(T \cup \{i\}) - v(T)$$

for all $S \subset T \subset N \setminus \{i\}$ and all $i \in N$, is violated for $S = \{3\}, T = \{2, 3\}$ and $i = 1$.

Lemma 2.3.3 describes a relation between pivot indices of two coalitions. It tells us that all agents preceding (with respect to the decreasing order by β) pivot $k(S)$, are also preceding pivot $k(T)$ when set S is extended to T.

Lemma 2.3.3. *Let $(N, e, d, \alpha) \in RA^N$. Let $S, T \in 2^N$ be such that $S \subset T$. Then*

$$k(S) \leq k(T).$$

Proof. Let $S \subset T \subset N$. In order to prove that $k(S) \leq k(T)$, we first assume that $D(T) = 0$. Then $k(T) = 1$ and

$$E \leq \sum_{j \in N \setminus T} d_j.$$

Hence,

$$E \leq \sum_{j \in N \setminus T} d_j + \sum_{j \in T \setminus S} d_j = \sum_{j \in N \setminus S} d_j.$$

This implies $D(S) = 0$ and $k(S) = 1$.

Next assume that $D(T) > 0$. First note that

$$\begin{aligned} D(S) &= \max\left\{0, E - \sum_{j\in N\setminus S} d_j\right\} \\ &= \max\left\{0, E - \sum_{j\in N\setminus T} d_j - \sum_{j\in T\setminus S} d_j\right\} \\ &= \max\left\{0, D(T) - \sum_{j\in T\setminus S} d_j\right\}. \end{aligned}$$

If $D(S) = 0$, then $k(S) = 1 \leq k(T)$ by definition. Therefore, we may assume that $D(S) > 0$ and $D(S) = D(T) - \sum_{j\in T\setminus S} d_j$. It follows that

$$\begin{aligned} \sum_{l=1}^{k(S)-1} d(T_l) &= \sum_{l=1}^{k(S)-1} d(S_l) + \sum_{l=1}^{k(S)-1} d(T_l \setminus S_l) \\ &\leq D(S) + \sum_{l=1}^{k(S)-1} d(T_l \setminus S_l) \\ &\leq D(S) + \sum_{j\in T\setminus S} d_j \\ &= D(T) \\ &< \sum_{l=1}^{k(T)} d(T_l) \end{aligned}$$

where the first and last inequality follows from (2.6). This implies that $k(S) - 1 < k(T)$ and consequently $k(S) \leq k(T)$. □

Lemma 2.3.4 provides a monotonicity result for optimal assignments. Hence, each set of agents S_l, with $l \in \{1, \dots, m\}$ obtains at most as much resources in the optimal assignment x^S of a coalition S, as it obtains in the optimal assignment x^T of a larger coalition T.

Lemma 2.3.4. *Let $(N, e, d, \alpha) \in RA^N$. Let $S, T \in 2^N$ be such that $S \subset T$. Let $x^T \in X(T, D(T), d|_T, \alpha|_T)$ and $x^S \in X(S, D(S), d|_S, \alpha|_S)$. Then*

$$\sum_{j\in S_l} x_j^S \leq \sum_{j\in S_l} x_j^T$$

for all $l \in \{1, \dots, m\}$.

Proof. Let $l \in \{1, \dots, m\}$. If $D(S) = 0$, then it holds that

$$\sum_{j \in S_l} x_j^S = 0 \leq \sum_{j \in S_l} x_j^T.$$

Assume $D(S) > 0$. Then $D(T) > 0$ and, as before in the proof of Lemma 2.3.3, this implies that $D(S) = D(T) - \sum_{j \in T \setminus S} d_j$. If $l > k(S)$, then $x_j^S = 0$ for all $j \in S_l$ and

$$\sum_{j \in S_l} x_j^S = 0 \leq \sum_{j \in S_l} x_j^T.$$

If $l < k(T)$, then it holds that $x_j^T = d_j$ for all $j \in T_l$. Hence, since $S_l \subset T_l$,

$$\sum_{j \in S_l} x_j^S \leq \sum_{j \in S_l} d_j = \sum_{j \in S_l} x_j^T.$$

Since $S \subset T$, it follows from Lemma 2.3.3 that $k(S) \leq k(T)$. Therefore, the only case that remains to be considered is $l = k(S) = k(T)$. In that case

$$\begin{aligned}
\sum_{j \in S_l} x_j^S &= D(S) - \sum_{t=1}^{l-1} d(S_t) \\
&\leq D(S) - \sum_{t=1}^{l-1} d(S_t) + d(T_l \setminus S_l) - \sum_{j \in T_l \setminus S_l} x_j^T + \sum_{t=l+1}^{m} d(T_t \setminus S_t) \\
&= D(S) - \sum_{t=1}^{l-1} d(S_t) - \sum_{t=1}^{l-1} d(T_t \setminus S_t) + \sum_{j \in T \setminus S} d_j - \sum_{j \in T_l \setminus S_l} x_j^T \\
&= D(S) + \sum_{j \in T \setminus S} d_j - \sum_{t=1}^{l-1} d(T_t) - \sum_{j \in T_l \setminus S_l} x_j^T \\
&= D(T) - \sum_{t=1}^{l-1} d(T_t) - \sum_{j \in T_l \setminus S_l} x_j^T \\
&= \sum_{j \in T_l} x_j^T - \sum_{j \in T_l \setminus S_l} x_j^T \\
&= \sum_{j \in S_l} x_j^T
\end{aligned}$$

where the first and fifth equality hold by Theorem 2.2.1. $\square$

A game $v \in TU^N$ is called *balanced* if the *core* $C(v)$ of the game is non-empty. The core of a game consists of those efficient allocations such that no coalition has an incentive to split off from the grand coalition, i.e.

$$C(v) = \left\{ y \in \mathbb{R}^N \middle| \sum_{j \in N} y_j = v(N), \sum_{j \in S} y_j \geq v(S) \text{ for all } S \in 2^N \right\}.$$

Theorem 2.3.5. *Let $(N, e, d, \alpha) \in RA^N$ with corresponding RA-game v^R and let $x^N \in X(N, E, d, \alpha)$. Let $y = (\alpha_i x_i^N)_{i \in N}$ be the direct allocation. Then $y \in C(v^R)$.*

Proof. First note that $\sum_{j \in N} y_j = v^R(N)$ by definition. Secondly, consider $S \subset N$. Then

$$\begin{aligned}
\sum_{j \in S} y_j &= \sum_{j \in S} \alpha_j x_j^N \\
&= \sum_{l=1}^{k-1} \beta_l d(S_l) + \beta_k \sum_{j \in S_k} x_j^N \\
&\geq \sum_{l=1}^{k(S)-1} \beta_l d(S_l) + \beta_{k(S)} \sum_{j \in S_{k(S)}} x_j^S \\
&= \sum_{l=1}^{k(S)-1} \beta_l d(S_l) + \beta_{k(S)} \left(D(S) - \sum_{l=1}^{k(S)-1} d(S_l) \right) \\
&= v^R(S).
\end{aligned}$$

The second and third equality follows from Theorem 2.2.1. From Lemma 2.3.3 we see that $k(S) \leq k$. If $k(S) < k$, then the inequality is readily verified. If $k(S) = k$, then we use Lemma 2.3.4 to see that this inequality holds. The last equality follows from Theorem 2.3.2. $\square$

By Theorem 2.3.5 it follows that RA-games are balanced. In fact, by Corollary 2.3.1 it follows that every RA-game is totally balanced.

Corollary 2.3.6. *Every RA-game is totally balanced.*

2.4 Compromise Stability and the Nucleolus

In this section we prove that RA-games are compromise stable and we derive an explicit expression for the nucleolus of RA-games.

Quant et al. (2005) defined a game $v \in TU^N$ to be *compromise stable* if

$$C(v) = CC(v)$$

and $CC(v) \neq \emptyset$. Here the *core cover* $CC(v)$, as introduced by Tijs and Lipperts (1982), is given by

$$CC(v) = \left\{ y \in \mathbb{R}^N \,\middle|\, \sum_{j \in N} y_j = v(N), m(v) \leq y \leq M(v) \right\}$$

where the *utopia demand* $M_i(v)$ of agent $i \in N$ is defined by

$$M_i(v) = v(N) - v(N \backslash \{i\})$$

and the *minimum right* $m_i(v)$ of agent $i \in N$ equals

$$m_i(v) = \max_{S: S \ni i} \left\{ v(S) - \sum_{j \in S \backslash \{i\}} M_j(v) \right\}.$$

Note that $m_i(v) \geq v(\{i\})$ for all $i \in N$. Moreover, for any TU-game v, $C(v) \subset CC(v)$ holds. Thus, the core cover equals the set of all efficient allocation vectors such that players receive at least their minimum right but no more than their utopia demand.

Moreover, it was proved by Quant et al. (2005) that each BR-game is both convex and compromise stable while reversely, each convex and compromise stable game is strategically equivalent[2] to a BR-game.

In what follows N_{-i} is a shorthand notation for $N \setminus \{i\}$.

Lemma 2.4.1. *Let* $(N, e, d, \alpha) \in RA^N$*, let* v^R *be the corresponding RA-game, and let* $i \in N$*.*

---

[2]Two TU-games v and w with player set N are called strategically equivalent if there exist a positive real number k and a vector $a \in \mathbb{R}^N$ such that $w(S) = kv(S) + \sum_{j \in S} a_j$ for all $S \in 2^N$.

1. If $d_i \geq E$*, then* $M_i(v^R) = v^R(N)$.

2. If $d_i < E$ *, then*

a. $M_i(v^R) = \alpha_i d_i$ *if* $i \in P_1$;

b. $M_i(v^R) = \beta_{k(N)} d_i$ *if* $i \in P_2 \cup P_3$ *and* $k(N_{-i}) = k(N)$;

c. $$M_i(v^R) = \beta_{k(N_{-i})} \left(\sum_{l=1}^{k(N_{-i})} d(N_l) - E + d_i \right) + \sum_{l=k(N_{-i})+1}^{k(N)-1} \beta_l d(N_l) + \beta_{k(N)} \left(E - \sum_{l=1}^{k(N)-1} d(N_l) \right) \quad \textit{if } i \in P_2 \cup P_3 \textit{ and } k(N_{-i}) < k(N).$$

Proof. *1*: Let $d_i \geq E$. Then $D(N_{-i}) = 0$ and $v^R(N_{-i}) = 0$. Therefore,

$$M_i(v^R) = v^R(N) - v^R(N_{-i}) = v^R(N).$$

2: Let $d_i < E$. Set $k = k(N)$,
2a: Let $i \in P_1$. We start by showing that $k(N_{-i}) = k$. By definition of k we have

$$\sum_{l=1}^{k-1} d(N_l) - d_i \leq E - d_i < \sum_{l=1}^{k} d(N_l) - d_i.$$

Since $i \in P_1 = N_1 \cup \cdots \cup N_{k-1}$, this implies

$$\sum_{l=1}^{k-1} d\left((N_{-i})_l\right) \leq D(N_{-i}) < \sum_{l=1}^{k} d\left((N_{-i})_l\right).$$

Consequently, $k(N_{-i}) = k$. Hence,

$$\begin{aligned} v^R(N_{-i}) &= \sum_{l=1}^{k-1} \beta_l d\left((N_{-i})_l\right) + \beta_k \left(D(N_{-i}) - \sum_{l=1}^{k-1} d\left((N_{-i})_l\right) \right) \\ &= \sum_{l=1}^{k-1} \beta_l d(N_l) - \alpha_i d_i + \beta_k \left(E - d_i - \sum_{l=1}^{k-1} d(N_l) + d_i \right) \\ &= v^R(N) - \alpha_i d_i. \end{aligned}$$

and

$$M_i(v^R) = v^R(N) - v^R(N_{-i}) = \alpha_i d_i.$$

2b/2c: Let $i \in P_2 \cup P_3$. By Theorem 2.3.2 it holds that

$$\begin{aligned} v^R(N_{-i}) &= \sum_{l=1}^{k(N_{-i})-1} \beta_l d\left((N_{-i})_l\right) + \beta_{k(N_{-i})} \left(D(N_{-i}) - \sum_{l=1}^{k(N_{-i})-1} d\left((N_{-i})_l\right) \right) \\ &= \sum_{l=1}^{k(N_{-i})-1} \beta_l d(N_l) + \beta_{k(N_{-i})} \left(E - d_i - \sum_{l=1}^{k(N_{-i})-1} d(N_l) \right) \end{aligned}$$

while

$$v^R(N) = \sum_{l=1}^{k-1} \beta_l d(N_l) + \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l) \right).$$

Then

$$\begin{aligned} M_i(v^R) &= v^R(N) - v^R(N_{-i}) \\ &= \sum_{l=k(N_{-i})}^{k-1} \beta_l d(N_l) - \beta_{k(N_{-i})} \left(E - d_i - \sum_{l=1}^{k(N_{-i})-1} d(N_l) \right) \\ &\quad + \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l) \right). \end{aligned}$$

If $k(N_{-i}) = k$, then this simplifies into

$$M_i(v^R) = \beta_k d_i.$$

If $k(N_{-i}) < k$, then

$$\begin{aligned} M_i(v^R) &= \beta_{k(N_{-i})} \left(\sum_{l=1}^{k(N_{-i})} d(N_l) - E + d_i \right) + \sum_{l=k(N_{-i})+1}^{k-1} \beta_l d(N_l) \\ &\quad + \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l) \right). \end{aligned}$$

This finishes 2b/2c. □

$M(v^R)$ constitutes an upper bound for the amount the agents reasonable can demand by allocating $v^R(N)$ among the agents. Those agents, who renounce their demand to the resources in order to obtain a larger total joint reward, are compensated by those agents who do obtain resources in the optimal assignment. It seems fair that the value of these renounced resources, and hence the compensation, is determined by the reward parameters of these latter agents. This can be seen from Lemma 2.4.1 since the utopia demands of e.g. agents in P_3 are computed without using their own reward parameter α. Hereby, the utopia demand of an agent $i \in P_2 \cup P_3$ may exceed $\alpha_i d_i$ (if $d_i \leq E$). This is illustrated in the following example.

Example 2.4.1. Consider the RA-problem of Example 2.3.1 with $v^R \in RA^N$. The vector of utopia demands equals $M(v^R) = (5, 1, 6)$. Note that

$$M_1(v^R) = 5 > \alpha_1 d_1 = 3.$$

This agent 1 obtains at most 2 units of resources in the optimal assignment, which is worth $2\alpha_1 = 2$. The renounced unit of resources ($d_1 - 2 = 1$) is valued using the reward parameter of the agent who does receive his full demand, i.e. $1\alpha_3 = 3$. $\Diamond$

Quant et al. (2005) proved that a game $v \in TU^N$ is compromise stable if and only if

$$v(S) \leq \max\left\{\sum_{j \in S} m_j(v), v(N) - \sum_{j \in N \setminus S} M_j(v)\right\}$$

for all $S \in 2^N \setminus \{\emptyset\}$.

Theorem 2.4.2. *Every RA-game is compromise stable.*

Proof. Clearly, $m(v^R) \geq 0$ for an RA-game v^R. Let $S \in 2^N \setminus \{\emptyset\}$. According to Quant et al. (2005), it suffices to prove that

$$v^R(S) \leq \max\left\{0, v^R(N) - \sum_{j \in N \setminus S} M_j(v^R)\right\} \tag{2.7}$$

in order to establish compromise stability.

For $D(S) = 0$ it holds that $v^R(S) = 0$ and inequality (2.7) is easily verified.

Let $D(S) > 0$. This implies that $v^R(S) > 0$ and $d_i < E$ for all $i \in N \setminus S$. It remains to prove that

$$v^R(N) - \sum_{j \in N \setminus S} M_j(v^R) \geq v^R(S) \tag{2.8}$$

Set $k = k(N)$. First let $\bigcup_{l=k}^{m} (N_l \setminus S_l) = \emptyset$. This tells us that

$$D(S) = E - \sum_{j \in N \setminus S} d_j = E - \sum_{l=1}^{k-1} d(N_l \setminus S_l),$$

which implies that $k(S) = k$. Hence,

$$\begin{aligned}
v^R(N) - \sum_{j \in N \setminus S} M_j(v^R) &= v^R(N) - \sum_{l=1}^{k-1} \sum_{j \in N_l \setminus S_l} M_j(v^R) \\
&= \sum_{l=1}^{k-1} \beta_l d(N_l) + \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l) \right) \\
&\quad - \sum_{l=1}^{k-1} \beta_l d(N_l \setminus S_l) \\
&= \sum_{l=1}^{k-1} \beta_l d(S_l) + \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l) \right) \\
&= \sum_{l=1}^{k-1} \beta_l d(S_l) + \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l \setminus S_l) - \sum_{l=1}^{k-1} d(S_l) \right) \\
&= \sum_{l=1}^{k-1} \beta_l d(S_l) + \beta_k \left(D(S) - \sum_{l=1}^{k-1} d(S_l) \right) \\
&= v^R(S).
\end{aligned}$$

Here, the second and last equality follows from Theorem 2.3.2 and Lemma 2.4.1(2a).

Secondly, let $\bigcup_{l=k}^{m} (N_l \setminus S_l) \neq \emptyset$ and assume that $k(N_{-i}) = k$ for all $i \in \bigcup_{l=k}^{m} (N_l \setminus S_l)$. We will prove inequality (2.8) by showing

$$v^R(N) - \sum_{l=1}^{k-1} \sum_{j \in N_l \setminus S_l} M_j(v^R) - v^R(S) \geq \sum_{l=k}^{m} \sum_{j \in N_l \setminus S_l} M_j(v^R).$$

For this,

$$
\begin{aligned}
& v^R(N) - \sum_{l=1}^{k-1} \sum_{j \in N_l \setminus S_l} M_j(v^R) - v^R(S) \\
= \; & \sum_{l=1}^{k-1} \beta_l d(N_l) + \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l) \right) - \sum_{l=1}^{k-1} \beta_l d(N_l \setminus S_l) \\
& - \sum_{l=1}^{k(S)-1} \beta_l d(S_l) - \beta_{k(S)} \left(E - \sum_{j \in N \setminus S} d_j - \sum_{l=1}^{k(S)-1} d(S_l) \right) \\
= \; & \beta_{k(S)} \left(\sum_{l=1}^{k(S)} d(S_l) - E + \sum_{j \in N \setminus S} d_j \right) + \sum_{l=k(S)+1}^{k-1} \beta_l d(S_l) \\
& + \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l) \right) \\
\geq \; & \beta_k \left(\sum_{l=1}^{k(S)} d(S_l) - E + \sum_{j \in N \setminus S} d_j + \sum_{l=k(S)+1}^{k-1} d(S_l) + E - \sum_{l=1}^{k-1} d(N_l) \right) \\
= \; & \beta_k \left(\sum_{l=k}^{m} d(N_l \setminus S_l) \right) \\
= \; & \beta_k \left(\sum_{l=k}^{m} \sum_{j \in N_l \setminus S_l} d_j \right) \\
= \; & \sum_{l=k}^{m} \sum_{j \in N_l \setminus S_l} M_j(v^R).
\end{aligned}
$$

Here, the first equality follows from Theorem 2.3.2 and Lemma 2.4.1(2a). The inequality holds because $\beta_k < \beta_l$ for all $l \in \{k(S), \ldots, k-1\}$. The last equality follows from the fact that $M_i(v^R) = \beta_k d_i$ for all $i \in \bigcup_{l=k}^{m} (N_l \setminus S_l)$.

Thirdly, let $\bigcup_{l=k}^{m} (N_l \setminus S_l) \neq \emptyset$ and let agent $i^* \in \bigcup_{l=k}^{m} (N_l \setminus S_l)$ be such that $k(N_{-i^*}) < k$. Without loss of generality, assume that $d_{i^*} \geq d_j$ for all $j \in \bigcup_{l=k}^{m} (N_l \setminus S_l)$. We will prove (2.8) by showing

$$
v^R(N) - \sum_{l=1}^{k-1} \sum_{j \in N_l \setminus S_l} M_j(v^R) - v^R(S) - M_{i^*}(v^R) \geq \sum_{l=k}^{m} \sum_{j \in N_l \setminus S_l : j \neq i^*} M_j(v^R).
$$

For this, observe that

$$
\begin{aligned}
& v^R(N) - \sum_{l=1}^{k-1} \sum_{j \in N_l \setminus S_l} M_j(v^R) - v^R(S) - M_{i^*}(v^R) \\
= \; & \sum_{l=1}^{k-1} \beta_l d(N_l) + \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l) \right) - \sum_{l=1}^{k-1} \beta_l d(N_l \setminus S_l) - \sum_{l=1}^{k(S)-1} \beta_l d(S_l) \\
& -\beta_{k(S)} \left(E - \sum_{j \in N \setminus S} d_j - \sum_{l=1}^{k(S)-1} d(S_l) \right) - \beta_{k(N_{-i^*})} \left(\sum_{l=1}^{k(N_{-i^*})} d(N_l) - E + d_i^* \right) \\
& - \sum_{l=k(N_{-i^*})+1}^{k-1} \beta_l d(N_l) - \beta_k \left(E - \sum_{l=1}^{k-1} d(N_l) \right) \\
= \; & \beta_{k(S)} \left(\sum_{l=1}^{k(S)} d(S_l) - E + \sum_{j \in N \setminus S} d_j \right) + \sum_{l=k(S)+1}^{k-1} \beta_l d(S_l) \\
& -\beta_{k(N_{-i^*})} \left(\sum_{l=1}^{k(N_{-i^*})} d(N_l) - E + d_{i^*} \right) - \sum_{l=k(N_{-i^*})+1}^{k-1} \beta_l d(N_l) \\
= \; & \beta_{k(S)} \left(\sum_{l=1}^{k(S)} d(S_l) - E + \sum_{j \in N \setminus S} d_j \right) + \sum_{l=k(S)+1}^{k(N_{-i^*})-1} \beta_l d(S_l) \\
& +\beta_{k(N_{-i^*})} \left(d(S_{k(N_{-i^*})}) - \sum_{l=1}^{k(N_{-i^*})} d(N_l) + E - d_{i^*} \right) \\
& - \sum_{l=k(N_{-i^*})+1}^{k-1} \beta_l d(N_l \setminus S_l) \\
\geq \; & \beta_{k(N_{-i^*})} \left(\sum_{l=1}^{k(S)} d(S_l) - E + \sum_{j \in N \setminus S} d_j + \sum_{l=k(S)+1}^{k(N_{-i^*})-1} d(S_l) \right. \\
& \left. + d(S_{k(N_{-i^*})}) - \sum_{l=1}^{k(N_{-i^*})} d(N_l) + E - d_{i^*} - \sum_{l=k(N_{-i^*})+1}^{k-1} d(N_l \setminus S_l) \right) \\
= \; & \beta_{k(N_{-i^*})} \left(\sum_{l=k}^{m} \sum_{j \in N_l \setminus S_l} d_j - d_{i^*} \right)
\end{aligned}
$$

$$
\begin{aligned}
&= \beta_{k(N_{-i^*})}\left(\sum_{l=k}^{m}\left(\sum_{\substack{j\in N_l\setminus S_l:\\ k(N_{-j})=k}} d_j + \sum_{\substack{j\in N_l\setminus S_l:\\ k(N_{-j})<k; j\neq i^*}} d_j\right)\right)\\
&\geq \sum_{l=k}^{m}\left(\sum_{\substack{j\in N_l\setminus S_l:\\ k(N_{-j})=k}} \beta_k d_j + \sum_{\substack{j\in N_l\setminus S_l:\\ k(N_{-j})<k; j\neq i^*}} \beta_{k(N_{-j})} d_j\right)\\
&= \sum_{l=k}^{m}\left(\sum_{\substack{j\in N_l\setminus S_l:\\ k(N_{-j})=k}} M_j(v^R) + \sum_{\substack{j\in N_l\setminus S_l:\\ k(N_{-j})<k; j\neq i^*}} \beta_{k(N_{-j})}\left(\sum_{q=1}^{k(N_{-j})} d(N_q) - E + d_j\right.\right.\\
&\quad \left.\left. + \sum_{q=k(N_{-j})+1}^{k-1} d(N_q) + E - \sum_{q=1}^{k-1} d(N_q)\right)\right)\\
&\geq \sum_{l=k}^{m}\left(\sum_{\substack{j\in N_l\setminus S_l:\\ k(N_{-j})=k}} M_j(v^R) + \sum_{\substack{j\in N_l\setminus S_l:\\ k(N_{-j})<k; j\neq i^*}} \left(\beta_{k(N_{-j})}\left(\sum_{q=1}^{k(N_{-j})} d(N_q) - E + d_j\right)\right.\right.\\
&\quad \left.\left. + \sum_{q=k(N_{-j})+1}^{k-1} \beta_q d(N_q) + \beta_k\left(E - \sum_{q=1}^{k-1} d(N_q)\right)\right)\right)\\
&= \sum_{l=k}^{m} \sum_{j\in N_l\setminus S_l: j\neq i^*} M_j(v^R).
\end{aligned}
$$

The first equality follows from Theorem 2.3.2 and Lemma 2.4.1(2a,2b). The third equality holds by using Lemma 2.3.3 such that $k(N_{-i^*}) \geq k(S)$ for $i^* \in N \setminus S$. The first and second inequality are due to the fact that $\beta_{k(N_{-i^*})} < \beta_l$ for all $l \in \{k(S), \ldots, k(N_{-i^*}) - 1\}$, $\beta_{k(N_{-i^*})} > \beta_l$ for all $l \in \{k(N_{-i^*}) + 1, \ldots, k-1\}$, and $\beta_{k(N_{-i^*})} \geq \beta_{k(N_{-j})}$ for all $j \in \bigcup_{l=k}^{m}(N_l \setminus S_l)$. For this note that for all $j \in \bigcup_{l=k}^{m}(N_l \setminus S_l)$, $\beta_{k(N_{-i^*})} \geq \beta_{k(N_{-j})}$ by the fact that $d_{i^*} \geq d_j$. $\square$

Next we derive an explicit expression for a specific core element of an RA-game: the nucleolus (Schmeidler (1969)). Recall (cf. Aumann and Maschler (1985)) that the nucleolus $n(v_{E,d})$ of a bankruptcy game $v_{E,d} \in TU^N$ can be computed as follows:

$$n(v_{E,d}) = AM(E,d) = \begin{cases} CEA(E, \frac{1}{2}d) & \text{if } \sum\limits_{j\in N} d_j \geq 2E; \\ d - CEA\left(\sum\limits_{j\in N} d_j - E, \frac{1}{2}d\right) & \text{if } \sum\limits_{j\in N} d_j < 2E; \end{cases}$$

where

$$CEA(\tilde{E}, \tilde{d}) = (\min\{\lambda, \tilde{d}_i\})_{i\in N}$$

with λ such that $\sum_{j\in N} \min\{\lambda, \tilde{d}_j\} = \tilde{E}$.

Although v^R is not necessarily convex and therefore not strategically equivalent to a BR-game (ref. Quant et al. (2005)), the nucleolus of v^R corresponds to the AM-rule of a related bankruptcy problem.

Theorem 2.4.3. *Let* $(N, e, d, \alpha) \in RA^N$ *and let* v^R *be the corresponding RA-game. Then*

$$n(v^R) = AM\left(v^R(N), M(v^R)\right).$$

Proof. Set $\hat{E} = v^R(N)$ and $\hat{d} = M(v^R)$. Observe that

$$\sum_{j\in N} \hat{d}_j = \sum_{j\in N} M_j(v^R) \geq v^R(N) = \hat{E}$$

because $C(v^R) \neq \emptyset$ and the core is a subset of the core cover for any game. Hence $(\hat{E}, \hat{d})$ is a bankruptcy problem and, consequently

$$n(v_{\hat{E},\hat{d}}) = AM(\hat{E}, \hat{d}).$$

Next we show that $C(v^R) = C(v_{\hat{E},\hat{d}})$. Observe that $v^R(N) = v_{\hat{E},\hat{d}}(N)$. Since RA-games satisfy inequality (2.7), which implies that

$$v^R(S) \leq v_{\hat{E},\hat{d}}(S)$$

for all $S \in 2^N \backslash \{\emptyset\}$, it is clear that $C(v_{\hat{E},\hat{d}}) \subset C(v^R)$. To prove also the reverse inclusion, let $y \in C(v^R)$ and $S \subset N$. We will prove that $\sum_{j\in S} y_j \geq v_{\hat{E},\hat{d}}(S)$. Since

$$\sum_{j\in S} y_j \geq \sum_{j\in S} v^R(\{i\}) \geq 0$$

and

$$\begin{aligned}\sum_{j\in S} y_j &= v^R(N) - \sum_{j\in N\setminus S} y_j \\ &\geq v^R(N) - \sum_{j\in N\setminus S} M_j(v^R).\end{aligned}$$

we find that

$$\sum_{j\in S} y_j \geq v_{\hat{E},\hat{d}}(S).$$

Potters and Tijs (1994) proved that for any two games $v, w \in TU^N$ with $C(v) = C(w)$ and v convex, $n(v) = n(w)$ holds. From this we may conclude that

$$n(v^R) = n(v_{\hat{E},\hat{d}})$$

and hence

$$n(v^R) = AM(\hat{E}, \hat{d}) = AM\left(v^R(N), M(v^R)\right).$$

$\square$

Example 2.4.2. Consider the RA-problem of Example 2.3.1 with $v^R \in RA^N$. The nucleolus of game v^R equals $n(v^R) = AM\,(8, (5, 1, 6)) = (3\frac{1}{4}, \frac{1}{2}, 4\frac{1}{4})$. $\Diamond$

Chapter 3

Resource Allocation Problems with Concave Reward Functions

3.1 Introduction

In this chapter, which is based on Grundel et al. (2013b), we analyze a resource allocation model with a common-pool resource in which the sum of the claims of all agents exceeds the total amount of resources. Young (1995) introduced a general framework for the "type" of a claimant: "the type of a claimant is a complete description of the claimant for purposes of the allocation, and determines the extent of a claimant's entitlement to the good". In our model we assume that the claim represents the maximum of resources an agent can handle. Therefore, an agent is never assigned more than this claim. Furthermore, we characterize each agent by an individual strictly increasing, continuous, and concave monetary reward function which allows for monetary compensations among agents, given a certain assignment of resources. This chapter generalizes the model in Chapter 2 where resource allocation problems of this type are considered for agents with linear reward functions.

Our model is applicable for various kinds of common-pool resource problems. For example, consider a common-pool of water, which should be distributed among a farmer, a large-scale horticultural company and a factory.

There is insufficient water to meet the rightful claims of all agents. The possibility of compensating agents who cede water to others monetarily, allows agents to search for acceptable and fair alternatives. Agents who do not obtain their claim can use this compensation for possible alternatives to using water. If an agent requires water more urgently than others at a certain level of assigned water, then this is incorporated in the model by appropriate concavity requirements in the reward functions.

Sustainable exploitation of common-pool natural resources, such as water, requires cooperation among users (Ostrom et al. (1994)). In practice, agents coordinate water extraction through various arrangements. They specify the assignment of water and compensation through monetary transfers (Ostrom et al. (1994) and Dinar (2007)). The economic literature includes several papers that focus on various aspects of international water sharing issues and their stability in a basin setting (Ambec and Sprumont (2002), Ambec and Ehlers (2008), Wang (2011), Weikard et al. (2013), Van den Brink et al. (2012)). For specific issues in resolutions in water resource management, we refer to Dinar (2004). Water resource issues have not only been modeled using cooperative game theory (see Parrachino et al. (2006) for an overview), but also using non-cooperative game theory (see Harris and Townsend (1981), Myerson (1979) for models with incomplete information and Pálvölgyi et al. (2010), Carraro et al. (2005), Condorelli (2013) for models with complete information).

In analyzing resource allocation problems, an assignment of resources is called optimal if the total joint monetary reward is maximized. It is shown that an assignment is optimal if and only if there does not exist a pair of agents for whom the sum of rewards increases by transferring resources from one agent to another. We show, by means of an example, how this characterization can be used to check optimality of an assignment. Then we apply cooperative game theory in order to allocate the corresponding maximal total joint reward in an adequate and fair way among the agents. In particular, we introduce a new class of transferable utility games, which is inspired by bankruptcy games (O'Neill (1982)). For these *resource allocation games* the value of a particular coalition reflects the maximum total joint reward that can be derived from the resources not claimed by the agents outside the coalition. We show that these games allow for core allocations which are stable against coalitional deviations. We analyze the nucleolus (Schmeidler (1969)) as a stable allocation rule and provide an explicit expression for the allocation prescribed by the nucleolus for a resource allocation game.

This chapter it is organized as follows. In Section 3.2 the formal model of resource allocation problems is provided and optimal assignments of resources are characterized. In Section 3.3, we introduce corresponding resource allocation games and show the existence of stable allocations and analyze the nucleolus of these games. The concluding remarks in Section 3.4 concern the relation between RA-problems with linear reward function, as provided in Chapter 2, and RA-problems with concave reward functions.

3.2 Resource Allocation Problems

This section formally introduces *resource allocation* (RA) *problems*, and characterizes optimal assignments of resources.

An RA-problem considers the assignment of resources among agents who have a claim on a common-pool resource. Let N represent the finite set of agents, $E \geq 0$ the total amount (estate) of resources which has to be divided among the agents, and $d \in (0, \infty)^N$ a vector of demands, where for $i \in N$, d_i represents agent i's claim on the estate. It is assumed that $\sum_{j \in N} d_j \geq E$. Furthermore, for each agent $i \in N$ there exists a *reward function* r_i on $[0, d_i]$ describing the monetary reward to agent i: for every $z \in [0, d_i], r_i(z)$ denotes the monetary reward for agent i if he is assigned z units of resource. In this chapter is assumed that for all $i \in N$, r_i is a strictly increasing, continuous, and concave reward function on $[0, d_i]$ with $r_i(0) = 0$. An RA-problem will be summarized by (N, E, d, r), with $r = \{r_i\}_{i \in N}$. The class of all RA-problems with set of agents N is denoted by RA^N.

Let $F(N, E, d, r)$ denote the set of assignments of resources given by

$$F(N, E, d, r) = \left\{ x \in \mathbb{R}^N \middle| \sum_{j \in N} x_j = E, x_i \in [0, d_i] \text{ for all } i \in N \right\}.$$

So, in an assignment, we assume that the complete estate E is assigned among the agents and that no agent can get more than its demand.

Throughout this article, assignments of resources which maximize the total joint monetary reward are considered. The remainder of this section is dedicated to characterizing these *optimal assignments*.

Let $(N, E, d, r) \in RA^N$. The maximum total joint monetary reward $v(N, E, d, r)$ is determined by

$$v(N,E,d,r) = \max\left\{\sum_{j\in N} r_j(x_j)\,\middle|\, x \in F(N,E,d,r)\right\}.$$

Note that this maximum exists due to the fact that $\sum_{j\in N} r_j$ is continuous on a compact domain. Furthermore, Lemma 3.2.1 shows that $v(N, E, d, r)$ is concave in the second coordinate E.

Lemma 3.2.1. *Let $(N, E, d, r) \in RA^N$. Then $v(N, E, d, r)$ is concave in E.*

Proof. Let $A, B \geq 0$ such that $\sum_{j\in N} d_j \geq A$ and $\sum_{j\in N} d_j \geq B$. We will prove that for all $\delta \in [0, 1]$ it holds that

$$\delta v(N,A,d,r) + (1-\delta)v(N,B,d,r) \leq v\left(N, \delta A + (1-\delta)B, d, r\right).$$

Let $x^A \in F(N, A, d, r)$ be such that $v(N, A, d, r) = \sum_{j\in N} r_j(x_j^A)$ and let $x^B \in F(N, B, d, r)$ be such that $v(N, B, d, r) = \sum_{j\in N} r_j(x_j^B)$. Then,

$$\begin{aligned}
& \delta v(N,A,d,r) + (1-\delta)v(N,B,d,r) \\
= \;& \delta \sum_{j\in N} r_j(x_j^A) + (1-\delta)\sum_{j\in N} r_j(x_j^B) \\
= \;& \sum_{j\in N}\left(\delta r_j(x_j^A) + (1-\delta) r_j(x_j^B)\right) \\
\leq \;& \sum_{j\in N} r_j\left(\delta x_j^A + (1-\delta)x_j^B\right) \\
\leq \;& \max\left\{\sum_{j\in N} r_j(x_j)\,\middle|\, x\in F(N,\delta A + (1-\delta)B, d, r)\right\} \\
= \;& v\left(N, \delta A + (1-\delta)B, d, r\right).
\end{aligned}$$

The first inequality follows from concavity of r_j. The second inequality is due to the fact that $(\delta x_i^A + (1-\delta)x_i^B)_{i\in N} \in F(N, \delta A + (1-\delta)B, d, r)$. □

The set $X(N,E,d,r)$ of optimal assignments is given by

$$X(N,E,d,r) = \left\{ x \in F(N,E,d,r) \middle| \sum_{j\in N} r_j(x_j) = v(N,E,d,r) \right\}.$$

The next theorem characterizes optimal assignments. It tells us that an assignment is optimal if and only if there does not exist a pair of agents for whom the sum of rewards increases by transferring resources from one agent to another.

Theorem 3.2.2. *Let $(N,E,d,r) \in RA^N$ and $x \in F(N,E,d,r)$. Then $x \in X(N,E,d,r)$ if and only if for all $i \in N$ with $x_i < d_i$ and for all $k \in N \setminus \{i\}$ with $x_k > 0$, there does not exist a positive $\epsilon \in (0, \min\{d_i - x_i, x_k\}]$ such that $r_i(x_i + \epsilon) + r_k(x_k - \epsilon) > r_i(x_i) + r_k(x_k)$.*[1]

Proof. We first prove the "only if" part. Let $x \in X(N,E,d,r)$. Suppose there exist an $i \in N$ with $x_i < d_i$, a $k \in N \setminus \{i\}$ with $x_k > 0$, and an $\epsilon \in (0, \min\{d_i - x_i, x_k\}]$ such that $r_i(x_i+\epsilon) + r_k(x_k - \epsilon) > r_i(x_i) + r_k(x_k)$. Now consider x' such that $x'_j = x_j$ for all $j \in N \setminus \{i,k\}$, $x'_i = x_i + \epsilon$, and $x'_k = x_k - \epsilon$. Note that $x' \in F(N,E,d,r)$ by construction of x' and definition of ϵ. Then

$$\begin{aligned}
\sum_{j\in N} r_j(x_j) &= r_i(x_i) + r_k(x_k) + \sum_{j \in N\setminus\{i,k\}} r_j(x_j) \\
&= r_i(x_i) + r_k(x_k) + \sum_{j \in N\setminus\{i,k\}} r_j(x'_j) \\
&< r_i(x_i+\epsilon) + r_k(x_k-\epsilon) + \sum_{j \in N\setminus\{i,k\}} r_j(x'_j) \\
&= \sum_{j\in N} r_j(x').
\end{aligned}$$

This establishes a contradiction with the optimality of x.

For the "if" part, let $x \in F(N,E,d,r)$ and $x \notin X(N,E,d,r)$. We will prove that there exists an $i \in N$ with $x_i < d_i$, a $k \in N \setminus \{i\}$ with $x_k > 0$, and an $\epsilon \in (0, \min\{d_i - x_i, x_k\}]$ such that $r_i(x_i+\epsilon) + r_k(x_k-\epsilon) > r_i(x_i) + r_k(x_k)$.

[1] If $(N,e,d,r) \in RA^N$ is such that for all $i \in N$, r_i is differentiable, then the conditions in Theorem 3.2.2 implies the necessary conditions that follow from the Kuhn-Tucker Theorem (Kuhn and Tucker (1951)) for an assignment to be optimal.

Let $x^N \in X(N, E, d, r)$. Clearly both sets $A_1 = \{i \in N | x_i^N > x_i\}$ and $A_2 = \{k \in N | x_k^N < x_k\}$ are nonempty. Note that for all $i \in A_1$ it holds that $x_i^N > 0$ and $x_i < d_i$. Vice versa, for all $k \in A_2$ it holds that $x_k > 0$ and $x_k^N < d_k$. The reward functions of agents $i \in A_1$ and $k \in A_2$ are outlined in Figure 3.1. By concavity of r it holds that, for all $i \in A_1$ and $\epsilon \in (0, x_i^N - x_i]$,

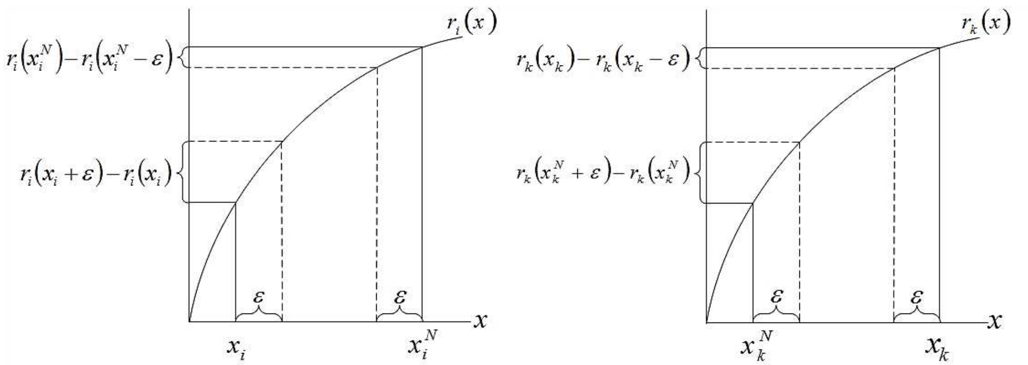

Figure 3.1: Reward functions of agents $i \in A_1$ and $k \in A_2$.

$$r_i(x_i + \epsilon) - r_i(x_i) \geq r_i(x_i^N) - r_i(x_i^N - \epsilon), \tag{3.1}$$

and, for all $k \in A_2$ and $\epsilon \in (0, x_k - x_k^N]$,

$$r_k(x_k^N + \epsilon) - r_k(x_k^N) \geq r_k(x_k) - r_k(x_k - \epsilon). \tag{3.2}$$

From the fact that $x^N \in X(N, E, d, r)$ it follows from the only if part that, for all $i \in A_1, k \in A_2$ and $\epsilon \in (0, \min\{x_i^N, d_k - x_k^N\}]$,

$$r_i(x_i^N) - r_i(x_i^N - \epsilon) \geq r_k(x_k^N + \epsilon) - r_k(x_k^N). \tag{3.3}$$

Since $(0, \min\{x_i^N - x_i, x_k - x_k^N\}] \subset (0, \min\{x_i^N, d_k - x_k^N\}]$, subsequently applying (3.1), (3.3), and (3.2) imply that, for all $i \in A_1$ and $k \in A_2$ and for all $\epsilon \in (0, \min\{x_i^N - x_i, x_k - x_k^N\}]$,

$$r_i(x_i + \epsilon) - r_i(x_i) \geq r_k(x_k) - r_k(x_k - \epsilon).$$

Suppose for all $i \in A_1, k \in A_2$ and $\epsilon \in (0, \min\{x_i^N - x_i, x_k - x_k^N\}]$ it holds that

$$r_i(x_i + \epsilon) - r_i(x_i) = r_k(x_k) - r_k(x_k - \epsilon).$$

Let $i \in A_1, k \in A_2$ and $\epsilon \in (0, \min\{x_i^N - x_i, x_k - x_k^N\}]$. Since inequality (3.1) is an equality now we have

$$r_i(x_i + \epsilon) - r_i(x_i) = r_i(x_i^N) - r_i(x_i^N - \epsilon).$$

By the fact that r_i is a strictly increasing, continuous, and concave function and $\epsilon > 0$ this tells us that r_i is linear on $[x_i, x_i^N]$. This is outlined in Figure 3.2. Similarly, we have an equality in (3.2) which implies that

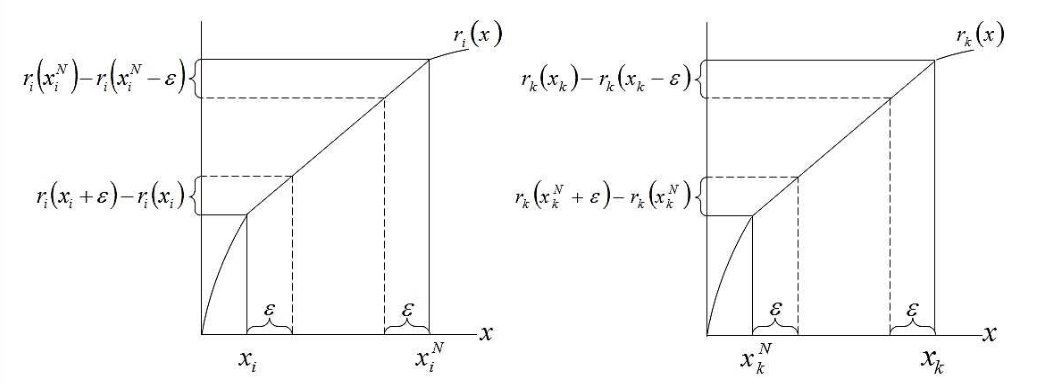

Figure 3.2: Linearity on $[x_i, x_i^N]$ and $[x_k^N, x_k]$.

$$r_k(x_k^N + \epsilon) - r_k(x_k^N) = r_k(x_k) - r_k(x_k - \epsilon)$$

which tells us that r_k is linear on $[x_k^N, x_k]$. Finally equality in (3.3) implies that

$$r_i(x_i^N) - r_i(x_i^N - \epsilon) = r_k(x_k^N + \epsilon) - r_k(x_k^N).$$

By linearity of r_i on $[x_i, x_i^N]$ and r_k on $[x_k^N, x_k]$ and the fact that the difference quotient of r_i on $[x_i, x_i^N]$ equals the difference quotient of r_k on $[x_k^N, x_k]$, it holds that

$$\frac{r_i(x_i^N) - r_i(x_i)}{x_i^N - x_i} = \frac{r_k(x_k) - r_k(x_k^N)}{x_k - x_k^N}.$$

Since this holds for all pairs of agents $i \in A_1$ and $k \in A_2$ it follows that

$$\frac{\sum_{j \in A_1} \left(r_j(x_j^N) - r_j(x_j)\right)}{\sum_{j \in A_1} \left(x_j^N - x_j\right)} = \frac{\sum_{j \in A_2} \left(r_j(x_j) - r_j(x_j^N)\right)}{\sum_{j \in A_2} \left(x_j - x_j^N\right)}.$$

As $x, x^N \in F(N,E,d,r)$ we have $\sum_{j\in N} x_j^N = \sum_{j\in N} x_j = E$ and, consequently, that $\sum_{j\in A_1}\left(x_j^N - x_j\right) = \sum_{j\in A_2}\left(x_j - x_j^N\right)$ which implies

$$\sum_{j\in A_1}\left(r_j(x_j^N) - r_j(x_j)\right) = \sum_{j\in A_2}\left(r_j(x_j) - r_j(x_j^N)\right). \tag{3.4}$$

Then

$$\begin{aligned}
\sum_{j\in N} r_j(x_j) &= \sum_{j\in A_1} r_j(x_j) + \sum_{j\in A_2} r_j(x_j) + \sum_{j\in N\setminus(A_1\cup A_2)} r_j(x_j)\\
&= \sum_{j\in A_1} r_j(x_j) + \sum_{j\in A_2} r_j(x_j) + \sum_{j\in N\setminus(A_1\cup A_2)} r_j(x_j^N)\\
&= \sum_{j\in A_1} r_j(x_j^N) + \sum_{j\in A_2} r_j(x_j^N) + \sum_{j\in N\setminus(A_1\cup A_2)} r_j(x_j^N)\\
&= \sum_{j\in N} r_j(x_j^N).
\end{aligned}$$

The second equality holds by the fact that for all $i \in N\setminus(A_1\cup A_2), x_i = x_i^N$, the third equality follows from (3.4).

This implies that $x \in X(N,E,d,r)$ which establishes a contradiction. Hence, there exists at least one pair of agents $i \in A_1, k \in A_2$ and $\epsilon \in (0, \min\{x_i^N - x_i, x_k - x_k^N\}]$, such that

$$r_i(x_i+\epsilon) - r_i(x_i) > r_k(x_k) - r_k(x_k-\epsilon).$$

Since $(0, \min\{x_i^N - x_i, x_k - x_k^N\}] \subset (0, \min\{d_i - x_i, x_k\}]$, this finishes the proof. $\square$

The following example illustrates how the conditions in Theorem 3.2.2 can be used in order to check optimality of an assignment of resources.

Example 3.2.1. Consider an RA-problem $(N,E,d,r) \in RA^N$ with $N = \{1,2,3,4\}$, estate $E = 7$, and vector of demands $d = (1,3,4,1)$. The reward functions of the agents are given by

$$\begin{aligned}
r_1(z) &= -3z^2 + 12z,\\
r_2(z) &= -z^2 + 6z,\\
r_3(z) &= \begin{cases} 2z, & \text{if } 0 \le z \le 2,\\ z+2, & \text{if } 2 < z \le 4,\end{cases}
\end{aligned}$$

$$r_4(z) = \frac{1}{2}z,$$

and are drawn in Figure 3.3. An optimal assignment equals $x = \big(1, 2\frac{1}{2}, 3\frac{1}{2}, 0\big)$

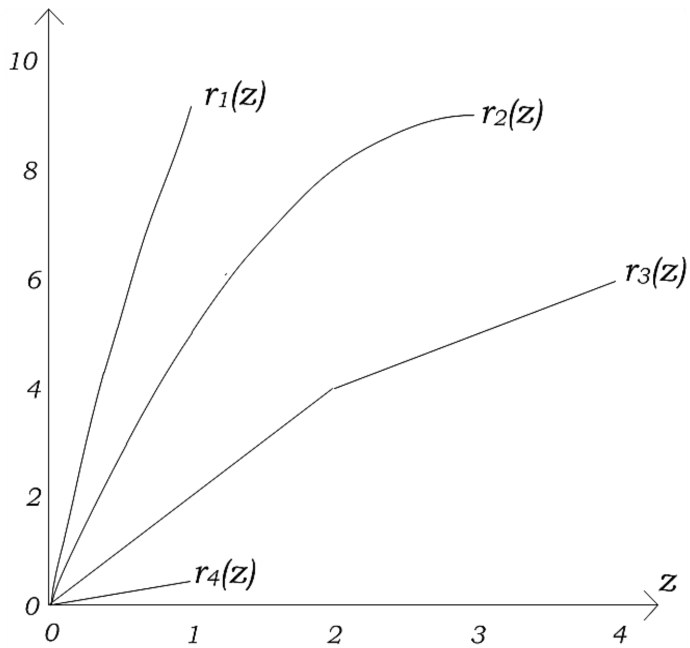

Figure 3.3: Reward functions $r_1(z)$, $r_2(z)$, $r_3(z)$, and $r_4(z)$.

with $r(x) = \big(9, 8\frac{3}{4}, 5\frac{1}{2}, 0\big)$, and total joint reward $23\frac{1}{4}$.

We can use Theorem 3.2.2 to check optimality of this assignment. For each pair of agents (i, k) it should hold for all $\epsilon \in (0, \min\{d_i - x_i, x_k\}]$ that $r_i(x_i + \epsilon) + r_k(x_k - \epsilon) \leq r_i(x_i) + r_k(x_k)$. From $\epsilon > 0$, $x_1 = d_1$, and $x_4 = 0$ it follows, respectively, that $i \neq 1$ and $k \neq 4$. Theorem 3.2.2 with $k = 1$ prescribes that for all $i \in \{2, 3, 4\}$ and all $\epsilon \in (0, \min\{d_i - x_i, 1\}]$, it should hold that $r_i(x_i + \epsilon) + r_1(1 - \epsilon) \leq r_i(x_i) + r_1(1)$ or equivalently, that

$$\frac{r_i(x_i + \epsilon) - r_i(x_i)}{\epsilon} \leq \frac{r_1(1) - r_1(1 - \epsilon)}{\epsilon}. \tag{3.5}$$

Inequality (3.5) holds since we know that for all $i \in \{2, 3, 4\}$ and $\epsilon \in (0, d_i - x_i]$

$$\frac{r_i(x_i + \epsilon) - r_i(x_i)}{\epsilon} \leq 1,$$

and for all $\epsilon \in (0, 1]$

$$\frac{r_1(1) - r_1(1 - \epsilon)}{\epsilon} \geq 6.$$

Theorem 3.2.2 with $k = 2$ and $i = 3$ prescribes that for all $\epsilon \in (0, \frac{1}{2}]$, it should hold that $r_3\left(3\frac{1}{2}+\epsilon\right)+r_2\left(2\frac{1}{2}-\epsilon\right) \leq r_3\left(3\frac{1}{2}\right)+r_2\left(2\frac{1}{2}\right)$ or equivalently, that

$$\frac{r_3\left(3\frac{1}{2}+\epsilon\right)-r_3\left(3\frac{1}{2}\right)}{\epsilon} \leq \frac{r_2\left(2\frac{1}{2}\right)-r_2\left(2\frac{1}{2}-\epsilon\right)}{\epsilon}.$$

This inequality is satisfied by concavity of r_2 and r_3 and the fact that $r_2'(2\frac{1}{2}) = r_3'(3\frac{1}{2}) = 1$. Hence,

$$\frac{r_3\left(3\frac{1}{2}+\epsilon\right)-r_3\left(3\frac{1}{2}\right)}{\epsilon} \leq r_3'\left(3\frac{1}{2}\right) = r_2'\left(2\frac{1}{2}\right) \leq \frac{r_2\left(2\frac{1}{2}\right)-r_2\left(2\frac{1}{2}-\epsilon\right)}{\epsilon}.$$

With $k = 2$ and $i = 4$ Theorem 3.2.2 prescribes that for all $\epsilon \in (0, 1]$, it should hold that

$$\frac{r_4(\epsilon)}{\epsilon} \leq \frac{r_2\left(2\frac{1}{2}\right)-r_2\left(2\frac{1}{2}-\epsilon\right)}{\epsilon}. \tag{3.6}$$

Inequality (3.6) hold since for all $\epsilon \in (0, 1]$

$$\frac{r_4(\epsilon)}{\epsilon} = \frac{1}{2},$$

and for all $\epsilon \in \left(0, 2\frac{1}{2}\right]$

$$\frac{r_2\left(2\frac{1}{2}\right)-r_2\left(2\frac{1}{2}-\epsilon\right)}{\epsilon} \geq r_2'\left(2\frac{1}{2}\right) = 1.$$

The check for optimality of x with $k = 3$ and $i = 2$ is analogous to $k = 2$ and $i = 3$, for $k = 3$ and $i = 4$ we use an argument similar to $k = 2$ and $i = 4$. $\Diamond$

Now consider a subgroup $S \subset N$. For a resource allocation problem $(N, E, d, r) \in RA^N$ the maximum total joint reward of a subgroup $S \subset N$ with $E' \leq E$ and $E' \leq \sum_{j\in S} d_j$ equals $v(S, E', d|_S, r|_S)$[2]. The next proposition shows that total maximization implies partial maximization.

Proposition 3.2.3. *Let $(N, E, d, r) \in RA^N$ and $x^N \in X(N, E, d, r)$. Then for all $S \subset N$ it holds that $(x_i^N)_{i\in S} \in X(S, \sum_{j\in S} x_j^N, d|_S, r|_S)$.*

[2] $d|_S \in \mathbb{R}^S$ denotes the restricted vector of demands for agents in S with respect to $d \in \mathbb{R}^N$; $r|_S$ refers to $\{r_j(z)\}_{j\in S}$

Proof. Since $x^N \in F(N,E,d,r)$ it holds that $(x_i^N)_{i\in S} \in F(S, \sum_{j\in S} x_j^N, d|_S, r|_S)$. Suppose there exists an $x^S \in F(S, \sum_{j\in S} x_j^N, d|_S, r|_S)$ such that $\sum_{j\in S} r_j(x_j^S) > \sum_{j\in S} r_j(x_j^N)$. Let $x \in \mathbb{R}^N$ be such that for all $i \in S : x_i = x_i^S$ and for all $i \in N \setminus S : x_i = x_i^N$. By the fact that for all $i \in N$ it holds that $x_i \in [0, d_i]$ and

$$\sum_{j\in N} x_i = \sum_{j\in S} x_j + \sum_{j\in N\setminus S} x_j = \sum_{j\in S} x_j^S + \sum_{j\in N\setminus S} x_j^N = \sum_{j\in S} x_j^N + \sum_{j\in N\setminus S} x_j^N = \sum_{j\in S} x_j^N = E$$

it follows that $x \in F(N,E,d,r)$. Now,

$$\sum_{j\in N} r_j(x_j) = \sum_{j\in S} r_j(x_j^S) + \sum_{j\in N\setminus S} r_j(x_j^N) > \sum_{j\in S} r_j(x_j^N) + \sum_{j\in N\setminus S} r_j(x_j^N) = \sum_{j\in N} r_j(x_j^N)$$

which establishes a contradiction with the fact that x^N is optimal. □

3.3 Resource Allocation Games

In this section we associate to each RA-problem a cooperative *resource allocation* (RA) *game*. A *transferable utility* (TU) *game* is an ordered pair (N, v) where N is the finite set of agents, and v the characteristic function on 2^N, the set of all subsets of N. The function v assigns to every coalition $S \in 2^N$ a real number $v(S)$ with $v(\emptyset) = 0$. Here, $v(S)$ is called the worth or value of coalition S. Here the coalitional value $v(S)$ is interpreted as the maximal total joint reward for coalition S when cooperating on its own. The values $v(S), S \in 2^N$, serve as reference points on the basis of which allocations of $v(N)$ are considered to be fair or stable. The set of all TU-games with set of agents N is denoted by TU^N. Where no confusion arises, we write v rather than (N, v).

Consider an RA-problem (N,E,d,r). We assume that a coalition S can only use the amount of resources $D(S)$ such that all agents outside S obtain resources up to their demand $d \in \mathbb{R}_+^N$. Let $(S, D(S), d|_S, r|_S) \in RA^S$ describe the associated resource allocation problem for S, where

$$D(S) = \max\left\{0, E - \sum_{j\in N\setminus S} d_j\right\}.$$

Note that $D(N) = E$. By the fact that $D(S) \leq \sum_{j\in S} d_j$ it follows that $(S, D(S), d|_S, r|_S)$ again is an RA-problem.

Corollary 3.3.1. *Let* $(N, E, d, r) \in RA^N$ *and let* $S \subset N$. *Then* $(S, D(S), d|_S, r|_S) \in RA^S$.

In the RA-game, v^R associated to an RA-problem (N, E, d, r), the worth of a coalition $S \in 2^N$ is defined as

$$v^R(S) = v\left(S, D(S), d|_S, r|_S\right).$$

Let $x^S \in X(S, D(S), d|_S, r|_S)$ be an optimal assignment of resources to agents in S. Clearly, $v^R(S)$ equals the total reward of agents in S associated to x^S. For simplicity, we write $v(S, D(S))$, rather than $v(S, D(S), d|_S, r|_S)$, $F(S, D(S))$, rather than $F(S, D(S), d|_S, r|_S)$, and $X(S, D(S))$, rather than $X(S, D(S), d|_S, r|_S)$.

A game $v \in TU^N$ is called *balanced* if the *core* $C(v)$ of the game is non-empty. The core of a game consists of those allocations of $v(N)$ such that no coalition has an incentive to split off from the grand coalition, i.e.

$$C(v) = \left\{ y \in \mathbb{R}^N \middle| \sum_{j \in N} y_j = v(N), \sum_{j \in S} y_j \geq v(S) \text{ for all } S \in 2^N \right\}.$$

Theorem 3.3.2. *Let* $(N, E, d, r) \in RA^N$ *with corresponding RA-game* $v^R \in TU^N$, *and choose* $x^N \in X(N, E)$. *Let* $y^N = (r_i(x_i^N))_{i \in N}$. *Then,* $y^N \in C(v^R)$.

Proof. First note that $\sum_{j \in N} y_j^N = v^R(N)$ by definition. Secondly, let $S \subset N$. Then,

$$\begin{aligned}
\sum_{j \in S} y_j^N &= \sum_{j \in S} r_j(x_j^N) \\
&= \max \left\{ \sum_{j \in S} r_j(x_j) \middle| x \in F\left(S, \sum_{j \in S} x_j^N\right) \right\} \\
&\geq \max \left\{ \sum_{j \in S} r_j(x_j) \middle| x \in F\left(S, D(S)\right) \right\} \\
&= v^R(S).
\end{aligned}$$

The second equality follows from Proposition 3.2.3. The inequality follows from the fact that $r_j(z)$ is increasing for all $i \in S$ and $\sum_{j \in S} x_j^N \geq D(S)$. This can be seen as follows.

$$\sum_{j \in S} x_j^N = \sum_{j \in N} x_j^N - \sum_{j \in N \setminus S} x_j^N = E - \sum_{j \in N \setminus S} x_j^N \geq E - \sum_{j \in N \setminus S} d_j.$$

because $x_i^N \leq d_i$ for all $i \in N$. Since, obviously $x_i^N \geq 0$ for all $i \in N$, also $\sum_{j\in S} x_j^N \geq 0$ and, consequently, $\sum_{j \in S} x_j^N \geq D(S)$. □

By Theorem 3.3.2 it follows that RA-games are balanced. Furthermore, by Corollary 3.3.1 it follows that every RA-game is totally balanced.

Corollary 3.3.3. *Every RA-game is totally balanced.*

The next lemma shows that RA-games satisfy some specific concavity conditions.

Lemma 3.3.4. *Let $(N, E, d, r) \in RA^N$ with corresponding RA-game $v^R \in TU^N$. Let $S, T, U \in 2^N$ be such that $S \subset T \subset N \setminus U$, $U \neq \emptyset$, and $v^R(S) > 0$. Then*

$$v^R(S \cup U) - v^R(S) \geq v^R(T \cup U) - v^R(T). \tag{3.7}$$

Proof. Let $x^T \in X(T, D(T))$ and $x^{T\cup U} \in X(T \cup U, D(T \cup U))$. Since $v^R(S) > 0$, we have $D(S) > 0$. Then

$$\begin{aligned}
&v^R(S \cup U) - v^R(S)\\
&= v\left(S \cup U, D(S \cup U)\right) - v(S, D(S))\\
&\overset{(1)}{=} v\left(S \cup U, D(S) + \sum_{j\in U} d_j\right) - v(S, D(S))\\
&= \max\left\{\sum_{j\in S\cup U} r_j(x_j) \,\middle|\, x \in F\left(S \cup U, D(S) + \sum_{j \in U} d_j\right)\right\} - v(S, D(S))\\
&\overset{(2)}{\geq} \max\left\{\sum_{j\in S\cup U} r_j(x_j) \,\middle|\, \{x_i\}_{i\in S} \in F\left(S, D(S) + \sum_{j\in U}(d_j - x_j^{T\cup U})\right),\right.\\
&\qquad \left.\{x_i\}_{i \in U} \in F\left(U, \sum_{j\in U} x_j^{T\cup U}\right)\right\} - v(S, D(S))\\
&= \max\left\{\sum_{j\in S} r_j(x_j) \,\middle|\, x \in F\left(S, D(S) + \sum_{j\in U}(d_j - x_j^{T\cup U})\right)\right\}\\
&\quad + \max\left\{\sum_{j\in U} r_j(x_j) \,\middle|\, x \in F\left(U, \sum_{j\in U} x_j^{T\cup U}\right)\right\} - v(S, D(S))
\end{aligned}$$

$$
\begin{aligned}
&= v\left(S, D(S) + \sum_{j\in U}(d_j - x_j^{T\cup U})\right) - v(S, D(S)) + v\left(U, \sum_{j\in U} x_j^{T\cup U}\right)\\
&\overset{(3)}{\geq} v\left(S, D(S) + \sum_{j\in U}(d_j - x_j^{T\cup U}) + \sum_{j\in T\setminus S}(d_j - x_j^{T\cup U})\right)\\
&\quad - v\left(S, D(S) + \sum_{j\in T\setminus S}(d_j - x_j^{T\cup U})\right) + v\left(U, \sum_{j\in U} x_j^{T\cup U}\right)\\
&\overset{(4)}{=} v\left(S, D(T\cup U) - \sum_{j\in T\cup U} x_j^{T\cup U} + \sum_{j\in S} x_j^{T\cup U}\right)\\
&\quad - v\left(S, D(S) + \sum_{j\in T\setminus S}(d_j - x_j^{T\cup U})\right) + v\left(U, \sum_{j\in U} x_j^{T\cup U}\right)\\
&\overset{(5)}{=} v\left(S, \sum_{j\in S} x_j^{T\cup U}\right) + v\left(U, \sum_{j\in U} x_j^{T\cup U}\right)\\
&\quad - v\left(S, D(S) + \sum_{j\in T\setminus S}(d_j - x_j^{T\cup U})\right)\\
&\overset{(6)}{=} \sum_{j\in S} r_j(x_j^{T\cup U}) + \sum_{j\in U} r_j(x_j^{T\cup U}) - v\left(S, D(S) + \sum_{j\in T\setminus S}(d_j - x_j^{T\cup U})\right)\\
&= \sum_{j\in S} r_j(x_j^{T\cup U}) + \sum_{j\in U} r_j(x_j^{T\cup U}) + \sum_{j\in T} r_j(x_j^{T\cup U}) - \sum_{j\in T} r_j(x_j^{T\cup U})\\
&\quad - v\left(S, D(S) + \sum_{j\in T\setminus S}(d_j - x_j^{T\cup U})\right)\\
&= \sum_{j\in T\cup U} r_j(x_j^{T\cup U}) - \sum_{j\in T\setminus S} r_j(x_j^{T\cup U}) - v\left(S, D(S) + \sum_{j\in T\setminus S}(d_j - x_j^{T\cup U})\right)\\
&\overset{(7)}{=} v\left(T\cup U, \sum_{j\in T\cup U} x_j^{T\cup U}\right) - v\left(T\setminus S, \sum_{j\in T\setminus S} x_j^{T\cup U}\right)\\
&\quad - v\left(S, D(S) + \sum_{j\in T\setminus S}(d_j - x_j^{T\cup U})\right)
\end{aligned}
$$

$$
\begin{aligned}
&= v\left(T \cup U, \sum_{j \in T \cup U} x_j^{T\cup U}\right) \\
&\quad - \max\left\{\sum_{j \in T\setminus S} r_j(x_j) \,\middle|\, x \in F\left(T \setminus S, \sum_{j \in T\setminus S} x_j^{T\cup U}\right)\right\} \\
&\quad - \max\left\{\sum_{j \in S} r_j(x_j) \,\middle|\, x \in F\left(S, D(S) + \sum_{j \in T\setminus S} (d_j - x_j^{T\cup U})\right)\right\} \\
&= v\left(T \cup U, \sum_{j \in T \cup U} x_j^{T\cup U}\right) \\
&\quad - \max\left\{\sum_{j \in T} r_j(x_j) \,\middle|\, \{x_i\}_{i \in T\setminus S} \in F\left(T \setminus S, \sum_{j \in T\setminus S} x_j^{T\cup U}\right),\right. \\
&\quad \left.\{x_i\}_{i \in S} \in F\left(S, D(S) + \sum_{j \in T\setminus S} (d_j - x_j^{T\cup U})\right)\right\} \\
&\overset{(8)}{\geq} v\left(T \cup U, \sum_{j \in T \cup U} x_j^{T\cup U}\right) \\
&\quad - \max\left\{\sum_{j \in T} r_j(x_j) \,\middle|\, x \in F\left(T, D(S) + \sum_{j \in T\setminus S} d_j\right)\right\} \\
&= v\left(T \cup U, \sum_{j \in T \cup U} x_j^{T\cup U}\right) - v\left(T, D(S) + \sum_{j \in T\setminus S} d_j\right) \\
&\overset{(9)}{=} v\left(T \cup U, D(T \cup U)\right) - v\left(T, D(T)\right) \\
&= v^R(T \cup U) - v^R(T)
\end{aligned}
$$

Equalities (1), (4), and (9) hold since $D(S) > 0$ respectively implies $D(S \cup U) = D(S) + \sum_{j\in U} d_j$, $D(T \cup U) = D(S) + \sum_{j\in U} d_j + \sum_{j \in T\setminus S} d_j$, and $D(T) = D(S) + \sum_{j\in T\setminus S} d_j$. Inequalities (2) and (8) follow by the fact that the maximum value decreases if an extra condition is involved in the optimization. Inequality (3) holds by Lemma 3.2.1. Equality (5) holds by the fact that $D(T \cup U) = \sum_{j \in T\cup U} x_j^{T\cup U}$. Equalities (6) and (7) follow from Proposition 3.2.3. $\square$

Example 3.3.1. Reconsider the RA-problem of Example 3.2.1. The corresponding values of $D(S)$, $X(S, D(S))$ and $v^R(S)$ are given in the table below.

S	1	2	3	4
$D(S)$	0	1	2	0
$x^S \in X(S, D(S))$	(0)	(1)	(2)	(0)
$v^R(S)$	0	5	4	0

S	1,2	1,3	1,4	2,3	2,4	3,4
$D(S)$	2	3	0	5	2	3
$x^S \in X(S, D(S))$	(1,1)	(1,2)	(0,0)	$(2\frac{1}{2}, 2\frac{1}{2})$	(2,0)	(3,0)
$v^R(S)$	14	13	0	$13\frac{1}{4}$	8	5

S	1,2,3	1,2,4	1,3,4	2,3,4	N
$D(S)$	6	3	4	6	7
$x^S \in X(S, D(S))$	$(1, 2\frac{1}{2}, 2\frac{1}{2})$	(1,2,0)	(1,3,0)	$(2\frac{1}{2}, 3\frac{1}{2}, 0)$	$(1, 2\frac{1}{2}, 3\frac{1}{2}, 0)$
$v^R(S)$	$22\frac{1}{4}$	17	14	$14\frac{1}{4}$	$23\frac{1}{4}$

Lemma 3.3.4 tells us that, e.g. $v(\{2,4\}) - v(\{2\}) \geq v(N) - v(\{1,2,3\})$. From $v(\{1,4\}) - v(\{4\}) < v(N) - v(\{2,3,4\})$, it follows that if $v(S) = 0$, inequality (3.7) may be violated. $\diamondsuit$

Now we derive an explicit expression for the nucleolus (cf. Schmeidler (1969)) of RA-games. For this we use some properties of *bankruptcy problems* and associated bankruptcy games. A *bankruptcy problem* is a triple (N, B, c), where N represents a finite set of agents, $B \geq 0$ is the estate which has to be divided among the agents, and $c \in [0, \infty)^N$ is a vector of claims, where for $i \in N$, c_i represents agent i's claim on the estate such that $\sum_{j \in N} c_j \geq B$. For the associated *bankruptcy* (BR) *game* $v_{B,c}$ the value of a coalition S is determined by the amount of B that is not claimed by agents in $N \setminus S$. Hence, for all $S \in 2^N$,

$$v_{B,c}(S) = \max\left\{0, B - \sum_{j \in N \setminus S} c_j\right\}.$$

Recall (cf. Aumann and Maschler (1985)) that the nucleolus $n(v_{E,d})$ of a bankruptcy game $v_{E,d} \in TU^N$ can be computed as follows:

$$n(v_{E,d}) = \begin{cases} CEA(E, \frac{1}{2}d) & \text{if } \sum\limits_{j \in N} d_j \geq 2E; \\ d - CEA\left(\sum\limits_{j \in N} d_j - E, \frac{1}{2}d\right) & \text{if } \sum\limits_{j \in N} d_j < 2E; \end{cases}$$

where

$$CEA(\tilde{E}, \tilde{d}) = (\min\{\lambda, \tilde{d}_i\})_{i \in N}$$

with λ such that $\sum_{j \in N} \min\{\lambda, \tilde{d}_j\} = \tilde{E}$.

It turns out that the nucleolus of an RA-game coincides with the nucleolus of an associated bankruptcy game.

Theorem 3.3.5. *Let $(N, E, d, r) \in RA^N$ and let v^R be the corresponding RA-game. Then*

$$n(v^R) = n(v_{B,c})$$

with $B = v^R(N)$ and $c = \left(v^R(N) - v^R(N \backslash \{i\})\right)_{i \in N}$.

Proof. Note that (N, B, c) is a BR-problem since $v^R(N) \geq 0$, and $C(v^R) \neq \emptyset$, implies that for $y \in C(v^R)$,

$$y_i = \sum_{j \in N} y_j - \sum_{j \in N \backslash \{i\}} y_j \leq v^R(N) - v^R(N \setminus \{i\}) = c_i, \tag{3.8}$$

and consequently, $\sum_{j \in N} c_j \geq \sum_{j \in N} y_j = v^R(N) = B$.

Next we show that $C(v^R) = C(v_{B,c})$. Clearly, $v^R(N) = B = v_{B,c}(N)$. First we will prove that $C(v_{B,c}) \subset C(v^R)$ by showing that for all $S \in 2^N, v^R(S) \leq v_{B,c}(S)$. Let $S \in 2^N$ and let $N \setminus S = \{i_1, \ldots, i_{|N \setminus S|}\}$. Without loss of generality we can assume that $v^R(S) > 0$. This implies that $D(S) > 0$ and, consequently, $D(N \setminus \{i_1\}) > 0, D(N \setminus \{i_1, i_2\}) > 0, \ldots,$ $D(N \setminus \{i_1, \ldots, i_{N \setminus S}\}) > 0$. Furthermore, $v^R(N \setminus \{i_1\}) > 0$, $v^R(N \setminus \{i_1, i_2\}) >$ $0, \ldots, v^R(N \setminus \{i_1, \ldots, i_{N \setminus S}\}) > 0$. For all $k \in \{0, \ldots, |N \setminus S| - 1\}$ we have by Lemma 3.3.4 that

$$v^R(N \setminus \{i_1, \ldots, i_k\}) - v^R(N \setminus \{i_1, \ldots, i_{k+1}\}) \geq v^R(N) - v^R(N \setminus \{i_{k+1}\}).$$

Since

$$\sum_{k=0}^{|N \setminus S|-1} \big(v^R(N \setminus \{i_1, \ldots, i_k\}) - v^R(N \setminus \{i_1, \ldots, i_{k+1}\})\big) = v^R(N) - v^R(S)$$

and

$$\sum_{k=0}^{|N \setminus S|-1} \big(v^R(N) - v^R(N \setminus \{i_{k+1}\})\big) = \sum_{j \in N \setminus S} \big(v^R(N) - v^R(N \setminus \{j\})\big) = \sum_{j \in N \setminus S} c_j,$$

we have that

$$v^R(N) - v^R(S) \geq \sum_{j \in N \setminus S} c_j.$$

Using $v^R(S) > 0$, this implies that

$$\begin{aligned} v^R(S) &\leq v^R(N) - \sum_{j \in N \setminus S} c_j \\ &= v_{B,c}(S). \end{aligned}$$

Secondly, in order to prove that $C(v^R) \subset C(v_{B,c})$, let $y \in C(v^R)$ and $S \in 2^N$. Since

$$\sum_{j \in S} y_j \geq \sum_{j \in S} v^R(\{j\}) \geq 0,$$

and, using (3.8),

$$\begin{aligned} \sum_{j \in S} y_j &= v^R(N) - \sum_{j \in N \setminus S} y_j \\ &\geq v^R(N) - \sum_{j \in N \setminus S} c_j \end{aligned}$$

we have that $\sum_{j \in S} y_j \geq v_{B,c}(S)$. It follows that $y \in C(v_{B,c})$.

Potters and Tijs (1994) proved that for any two games $v, w \in TU^N$ with $C(v) = C(w)$ and w convex, we have $n(v) = n(w)$. From the fact the $v_{B,c}$ is a BR-game, that BR-games are convex (Curiel et al. (1987)), and $C(v^R) = C(v_{B,c})$ we conclude that

$$n(v^R) = n(v_{B,c}).$$

□

Example 3.3.2. For the RA-game of Example 3.3.1, $v^R(N) = 23\frac{1}{4}$ and $\left(v^R(N) - v^R(N \setminus \{i\})\right)_{i \in N} = \left(9, 9\frac{1}{4}, 6\frac{1}{4}, 1\right)$. Theorem 3.3.5 implies that $n(v^R) = \left(9, 9\frac{1}{4}, 6\frac{1}{4}, 1\right) - CEA\left(N, 2\frac{1}{4}, \left(4\frac{1}{2}, 4\frac{5}{8}, 3\frac{1}{8}, \frac{1}{2}\right)\right) = \left(8\frac{5}{12}, 8\frac{2}{3}, 5\frac{2}{3}, \frac{1}{2}\right)$. $\Diamond$

3.4 Concluding Remarks

In the RA-problems of Chapter 3 now agents are characterized by a concave reward function rather than a linear reward function, as in Chapter 2. In fact, since linear functions fall within the class of concave functions, the model in Chapter 3 is a generalization of the model in Chapter 2.

RA-problems such as described in Chapter 2 are close to the existing model of BR-problems: agents are not only characterized by a claim, but also by a linear reward function. Therefore, an explicit comparison is made between properties of RA-games and BR-games. In Chapter 3 we consider concave reward functions in order to make RA-problems as such more consistent to problems in an economic setting. With respect to the results, the model of Chapter 2 allows for an explicit formula of the maximum total joint reward for each coalition of agents (Theorem 2.3.2), which is not the case in Chapter 3.

Chapter 4

River Sharing Allocation

4.1 Introduction

"Water is essential to life." This statement is often used as an opening sentence of papers relating to water management, in order to highlight the importance of this field of research. Sustainable exploitation of common-pool resources such as water requires cooperation among users (Ostrom et al. (1994)). However, managing water resource systems usually involves conflicts arising from social and political aspects, but also creates conflicts over sharing costs or benefits. These conflicts are mainly caused by asymmetric dependence on a water resource. Issues of water management involve allocation of water resources, groundwater management, water quality management, transboundary water disputes, and using water in electricity generation or irrigation. In practice, agents coordinate water extraction through various arrangements. They specify the assignment of water and compensation through monetary transfers (Ostrom et al. (1994) and Dinar (2007)).

This chapter deals with a simple example of a transboundary water conflict where an upstream user could withdraw such a large amount of water that downstream users are no longer able to withdraw their preferred amounts. In literature, this is known as the river sharing problem (Ambec and Sprumont (2002)). The river sharing problem deals with the fair distribution of the total welfare resulting from the optimal assignment of water among a set of agents along the river.

In order to tackle the river sharing problem, global institutes such as the

United Nations, came up with several multilateral agreements between states sharing a water resource. Two common principles for river sharing are absolute territorial sovereignty (ATS) and unlimited territorial integrity (UTI) (Salman (2007)). The ATS principle implies that a state could do whatever it wants with the inflow on its territory, irrespective to the harmful consequences this might have to downstream states. At the other end, the UTI principle states that the use of water resources within a state is permitted only in so far it does not cause damage or injury in the territory of other states. The principle of Territorial Integration of all Basin States (TIBS) combines both (extreme) principles by considering the problem in terms of benefits, rather than water. The TIBS principle requires the full sharing of both the benefits and costs of the management of an international watercourse such that it can be summarized in three steps: (1) the water rights over an international watercourse belong to all basin states combined, (2) the basin states are obliged to put the water from the international watercourse to its most productive use, and (3) each basin state has a right to a reasonable and equitable share of the benefit (wealth) that results from the optimal use of the water from the international watercourse. Observe that it is crucial for the implementation of the TIBS principle that the countries sharing a river have the possibility to make monetary transfers to each other. For an extended overview of the highlights in the law of international watercourses in the past century and the TIBS principle, we refer to Moes (2013).

For the river sharing problem Ambec and Sprumont (2002) introduce a model in which a group of agents is located along a single-stream river from upstream to downstream. Each agent is assumed to have quasi-linear preferences over river water and money. The benefits of consuming an amount of water are given by a strictly increasing, differentiable, and strictly concave reward function where satiation points do not exist. An assignment of the river water among the agents is called efficient when it maximizes the total joint reward. To sustain such an efficient assignment, agents can compensate each other by monetary transfers. This model has been tested in a realistic setting e.g. for the Jordan River by Jägerskog (2007) and the Nile by Dinar and Nigatu (2013). Moreover, the model has been expanded by adding more aspects. One of these aspects is allowing satiable agents (by Ambec and Ehlers (2008)) by the fact that overconsumption may cause flooding or increase sanitation costs with higher water extraction costs. Another aspect concerns the structure of the river. Khmelnitskaya (2010) considers rivers that have a sink-tree or rooted-tree structure allowing multiple springs or deltas. Van den Brink et al. (2012) combine the studies of rivers with multiple springs and satiable agents and suggest a new class of solutions based on the TIBS principle. We refer to Beal et al. (2013), for a recent survey of the river sharing problem.

Cooperative game theory has often been employed for analyzing river sharing allocation problems. Game theory provides insight into the fair allocation of the jointly generated maximal total joint reward of the utilization of river water. Kilgour and Dinar (2001) and Ambec and Sprumont (2002) were the first to model the river sharing problem as a cooperative TU-game on the set of participating agents. Ambec and Sprumont (2002) propose the downstream incremental solution, that is, the marginal vector of the game with the players ordered from upstream to downstream, as a solution to the problem of allocating the maximal total joint reward. For the purpose of climate change, Ambec and Dinar (2009) examine the robustness of such solutions to reduced water flow. Other solution concepts are upstream incremental solution, that is, the marginal vector of the game with players ordered from downstream to upstream, and the combination of both up- and downstream incremental solution (Van den Brink et al. (2007)), Harsanyi solutions (Van den Brink et al. (2003)), component efficient solutions (Van den Brink et al. (2007)), and the average tree solution (Herings et al. (2008)). Wang (2011) proposes a solution in which water trading is restricted to pairs of neighboring agents. Van den Brink et al. (2014) introduce a number of axioms to characterize these solutions. Parrachino et al. (2006) provide a review of various applications of cooperative game theory to issues of water resources.

In this chapter we modify the model of Ambec and Sprumont (2002) in the following way. We describe a single-stream river along which a number of agents are situated, each with a justified claim on the water. For each agent this demand is greater than or equal to the inflow on his territory. Moreover, each agent is characterized by a strictly increasing, differentiable, and concave reward function which describes the reward per unit of assigned water. Unlike the model of Ambec and Sprumont (2002) we incorporate satiation points by allowing each agent to extract water at most up to his demand. As the TIBS principle prescribes, we consider the assignments of water such that the total joint reward is maximized. We characterize optimal assignments of water for the group as a whole by means of the derivatives of the reward functions and present a methodology to determine such an assignment. Herein downstream transfer of water is possible, upstream transfer is not. For the corresponding river sharing allocation game, a coalition of players can distribute water assuming that players outside the coalition act in an individualistic way. Within a coalition, water can be distributed, taking into account that all in-between individualists, if possible, will extract water up to their demand. We show that the downstream incremental solution is in the core of the corresponding river sharing allocation game. We finish the chapter by presenting a solution that is directly based on the methodology to determine an optimal assignment of water.

This chapter is organized as follows. In Section 4.2 the formal model of river sharing allocation problems is provided and optimal assignments of water are characterized. In Section 4.3 we introduce corresponding river sharing allocation games and show the existence of stable allocations.

4.2 River Sharing Allocation Problems

This section formally introduces *river sharing allocation (RSA) problems.* After introducing the model, optimal assignments of water, in which the total joint reward is maximized, are characterized.

In an RSA-problem (N, e, d, r), $N = \{1, ..., n\}$ represents the set of agents on the river, numbered successively from upstream to downstream. Let $e \in \mathbb{R}^N_+$ is the vector of amounts of inflow of water. Here, e_i is the flow of water entering the river between agents $i-1$ and i, with e_1 the inflow before the most upstream agent 1. Agent i can extract water up to his demand d_i, where $d = (d_i)_{i \in N} \in \mathbb{R}^N_+$ is the corresponding vector of demands. Subtracting a higher amount of water is assumed to be infeasible. Furthermore, for each agent $i \in N$ there exists a *reward function* r_i on $[0, d_i]$ describing the monetary gain to agent i from assigned water: for every $z \in [0, d_i], r_i(z)$ denotes the reward for agent i if he is assigned z units of water. In this chapter it is assumed that for every $i \in N, r_i$ is a strictly increasing, differentiable, and concave reward function on $[0, d_i]$ with $r_i(0) = 0$. The class of RSA-problems with set of agents N is denoted by RSA^N.

The agents' objective is to maximize the total joint monetary reward of water use and they may increase these joint rewards by transferring water to others. Here, upstream agents may transfer their water to downstream agents. Water cannot be transported upstream.

We assume water scarcity by taking $e_i \leq d_i$ for all $i \in N$. In fact this can be done without loss of generality (cf. Ambec and Ehlers (2008)). Suppose (N, e, d, r) is such that there exist an agent $i \in N$ where $e_i > d_i$. By assumption it holds that agent i will never extract more water than d_i, and i will always transfer at least $e_i - d_i$. Now define $(N, \hat{e}, d, r)$, where $\hat{e}_i = d_i, \hat{e}_{i+1} = e_{i+1} + (e_i - d_i)$, and for all $j \in N \setminus \{i, i+1\}, \hat{e}_j = e_j$[1]. Now, the analysis of (N, e, d, r) and $(N, \hat{e}, d, r)$ coincide.

Let $F(N, e, d, r)$ be the set containing all efficient assignments such that for all $i \in N$ the sum of assigned water of all upstream agents of i (including i) does not exceed the available water (the total inflow of these upstream agents), the amount of assigned water is non-negative, and does not exceed demand d_i. Hence,

$$\begin{aligned} F(N, e, d, r) = & \Bigg\{ x \in \mathbb{R}^N \Bigg| \sum_{j \in N} x_j = \sum_{j \in N} e_j, \\ & \sum_{j=1}^{i} x_j \leq \sum_{j=1}^{i} e_j \text{ and } x_i \in [0, d_i] \text{ for all } i \in N \Bigg\}. \end{aligned}$$

Throughout this chapter, feasible assignments of water which maximize the total joint reward are considered. In the remainder of this chapter we refer to these assignments as *optimal assignments*.

Let $(N, e, d, r) \in RSA^N$. The maximum total joint reward $v(N, e, d, r)$ is

[1] If $i = n$, then we set $\hat{e} = ((e_j)_{j \in N \setminus \{n\}}, d_n)$, i.e. the amount $(e_n - d_n)$ is not extracted by any agent.

determined by

$$v(N,e,d,r) = \max\left\{\sum_{j\in N} r_j(x_j)\,\middle|\, x \in F(N,e,d,r)\right\}.$$

Note that this maximum exists due to the fact that $\sum_{j\in N} r_j$ is continuous on a compact domain. The set $X(N,e,d,r)$ of optimal assignments is given by

$$X(N,e,d,r) = \left\{x \in F(N,e,d,r)\,\middle|\, \sum_{j\in N} r_j(x_j) = v(N,e,d,r)\right\}.$$

Let $E_i(x)$ be the inflow of agent $i \in N$ for a feasible assignment $x \in F(N,e,d,r)$, i.e.

$$E_i(x) = e_i + \sum_{j=1}^{i-1} (e_j - x_j)\,.$$

Let $P(x)$ be the set of agents extracting all inflow entering their territory with respect to assignment x, i.e.

$$P(x) = \{p \in N | x_p = E_p(x)\}.$$

Without loss of generality, take $P(x) = \{p_1, \ldots, p_{m(x)}\}$, with $p_1 < p_2 < \cdots < p_{m(x)}$. Observe that by efficiency holds that $p_{m(x)} = n$. Define the partition $N = \{N_1(x), \ldots, N_{m(x)}(x)\}$, referred to as the *$x$-partition*, such that $N_1(x) = \{1, \ldots, p_1\}$ and for all $l \in \{2, \ldots, m(x)\}$, $N_l(x) = \bigcup_{j=p_{l-1}+1}^{p_l}\{j\}$. This implies that subsequent agents with a positive flow of water in-between are in the same element of the partition and the most downstream agent in a partition extracts all inflow, hence, is in $P(x)$.

The conditions in Theorem 4.2.1 consider the derivative $r_i'(z)$ for each $i \in N$. This derivative reflects the marginal contribution of an agent for extra assigned water. We provide the intuition behind the formal characterization of an optimal assignment in Theorem 4.2.1. An optimal assignment x leads to a partition of N. Within an element of such partition, the marginal contributions of agents coincide if both agents are assigned a positive amount of water which is lower than their demand. Indeed, if for such assignment the marginalities would not be equal, the joint reward can be increased by transferring water between players that have different marginalities. Agents who are assigned their demand have a marginal contribution greater than or equal to the other agents, since feasibility does not allow for any more water to be assigned. Vice versa, agents without assigned water have a marginal contribution lower than or equal to the other agents, since such agents are not able to transfer any water to others. Note that condition (1) implies for an optimal assignment $x \in X(N,e,d,r)$, that for two agents in the same partition, i.e. $i,k \in N_l(x)$ with $l \in \{1,\dots,m(x)\}$, $x_i \in (0,d_i)$ and $x_k \in (0,d_k)$ it holds that $r'_i(x_i) = r'_k(x_k)$. For two agents i,k in two different partitions where agent i is upstream of k, the marginal contribution of k is lower than or equal to the marginal contribution of i. If not, using a similar argument as inside one partition element, transferring water from agent i to k increases the total joint reward which contradicts the optimality.

Theorem 4.2.1. *Let $(N,e,d,r) \in RSA^N$ and $x \in F(N,e,d,r)$. Then $x \in X(N,e,d,r)$ if and only if the following two conditions are satisfied*

(1) $r'_i(x_i) \geq r'_k(x_k)$ for all $i,k \in N_l(x)$ with $l \in \{1,\dots,m(x)\}$ such that $x_i \in (0,d_i]$ and $x_k \in [0,d_k)$, and

(2) $r'_i(x_i) \geq r'_k(x_k)$ for all $i,k \in N$ such that $i < k, x_i \in (0,d_i]$ and $x_k \in [0,d_k)$.[2]

Proof. We first prove the "only if" part. Let $x \in X(N,e,d,r)$. We prove that if (1) or (2) is violated, there exists an assignment $\hat{x} \in F(N,e,d,r)$ such that $\sum_{j\in N} r_j(x_j) < \sum_{j\in N} r_j(\hat{x}_j)$.

Suppose (2) is violated for $i,k \in N$ where $i < k, x_i \in (0,d_i], x_k \in [0,d_k)$ such that $r'_i(x_i) < r'_k(x_k)$. This implies that there exists a $\delta > 0$ such that for all $\epsilon \in (0,\delta)$ it holds that $r_k(x_k+\epsilon) + r_i(x_i-\epsilon) > r_k(x_k) + r_i(x_i)$. From the fact that i is upstream of k it follows that there exists an $\epsilon \in (0,\delta)$ such that $\hat{x} \in F(N,e,d,r)$, where $\hat{x}$ is such that $\hat{x}_j = x_j$ for all $j \in N\backslash\{i,k\}$, $\hat{x}_k = x_k+\epsilon$,

[2]The conditions in Theorem 3.2.2 implies the necessary conditions that follow from the Kuhn-Tucker Theorem (Kuhn and Tucker (1951)) for an assignment to be optimal.

and $\hat{x}_i = x_i - \epsilon$. Now

$$\begin{aligned}
\sum_{j\in N} r_j(x_j) &= \sum_{j\in N\setminus\{i,k\}} r_j(x_j) + r_i(x_i) + r_k(x_k) && (4.1)\\
&= \sum_{j\in N\setminus\{i,k\}} r_j(\hat{x}_j) + r_i(x_i) + r_k(x_k) && (4.2)\\
&< \sum_{j\in N\setminus\{i,k\}} r_j(\hat{x}_j) + r_i(x_i - \epsilon) + r_k(x_k + \epsilon) && (4.3)\\
&= \sum_{j\in N} r_j(\hat{x}_j). && (4.4)
\end{aligned}$$

Hence, $x \notin X(N, e, d, r)$ which establishes a contradiction.

Now suppose (1) is violated for $N_l(x)$ with $l \in \{1, \ldots, m(x)\}$. Then there exists a pair of agents $i, k \in N_l(x)$ with $x_i \in (0, d_i], x_k \in [0, d_k)$ such that $r_i'(x_i) < r_k'(x_k)$. This tells us that there exists a $\delta > 0$ such that for all $\epsilon \in (0, \delta)$ it holds that $r_k(x_k + \epsilon) + r_i(x_i - \epsilon) > r_k(x_k) + r_i(x_i)$. From the fact that i and k both are in $N_l(x)$, and there is a positive flow in-between agents in $N_l(x)$ (i.e. for all $j \in N$ such that $\min\{i, k\} < j < \max\{i, k\}$ it holds that $x_j < E_j(x)$), there exists a feasible solution where water is transferred from agent i to k. Hence, there exists an $\epsilon \in (0, \delta)$ such that $\hat{x} \in F(N, e, d, r)$, where $\hat{x}$ is such that $\hat{x}_j = x_j$ for all $j \in N \setminus \{i, k\}, \hat{x}_k = x_k + \epsilon$, and $\hat{x}_i = x_i - \epsilon$. Similar to (4.1) - (4.4) it follows that $\hat{x} \notin X(N, e, d, r)$ which establishes a contradiction.

For the "if" part, let $x \in F(N, e, d, r)$, $x \notin X(N, e, d, r)$, and $x^N \in X(N, e, d, r)$. Clearly, the sets $A_1 = \{j \in N | x_j^N > x_j\}$, and $A_2 = \{j \in N | x_j^N < x_j\}$ both are non-empty. Note that for all $i \in A_1, x_i < d_i$ and $x_i^N > 0$. Vice versa, it holds for all $k \in A_2$ that $x_k^N < d_k$ and $x_k > 0$. From concavity it follows that

$$\begin{aligned}
r_i'(x_i^N) &\leq r_i'(x_i) \text{ for all } i \in A_1, \text{ and} && (4.5)\\
r_k'(x_k^N) &\geq r_k'(x_k) \text{ for all } k \in A_2. && (4.6)
\end{aligned}$$

Suppose for all $i \in A_1$ it holds that $r_i'(x_i^N) = r_i'(x_i)$, and for all $k \in A_2$ that $r_k'(x_k^N) = r_k'(x_k)$. By the fact that r is a strictly increasing and concave function it holds that for all $i \in A_1, r_i$ is linear on $[x_i, x_i^N]$ and for all $k \in A_2, r_k$ is linear on $[x_k^N, x_k]$. This can be verified using a similar argument as in the proof of Theorem 3.2.2 and outlined in Figure 3.2. Hereby, $\sum_{j \in N} r_j(x_j) = \sum_{j \in N} r_j(x_j^N)$ and hence $x \in X(N,e,d,r)$. This establishes a contradiction such that for at least one agent $i \in A_1$ it holds that $r_i'(x_i^N) < r_i'(x_i)$, or at least one $k \in A_2$ that $r_k'(x_k^N) > r_k'(x_k)$.

Let $\{N_1(x), \ldots, N_{m(x)}(x)\}$ be the x-partition and let $\{N_1(x^N), \ldots, N_{m(x^N)}(x^N)\}$ be the x^N-partition. Since, x and x^N do not coincide, these partitions could be different. The x-partition is outlined in Figure 4.1. Let $i \in A_1$ be such that $i \in N_l(x), i \in N_{\hat{l}}(x^N)$ and $r_i'(x_i^N) < r_i'(x_i)$. Since $x^N \in X(N,e,d,r)$ we know from the "only-if" part of the proof that assignment x^N satisfies (1) and (2). We will prove that either (1) or (2) is violated for assignment x.

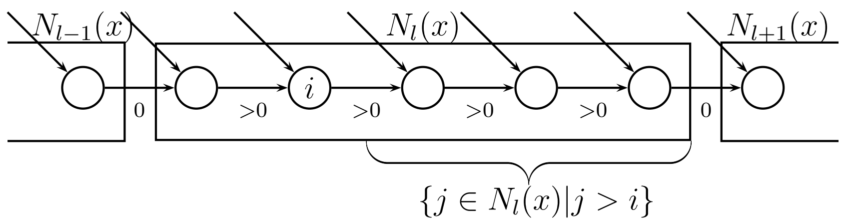

Figure 4.1: Agent $i \in N_l(x)$ in the x-partition.

First, assume $\{j \in N_l(x) \cap A_2 | j > i\} \neq \emptyset$ and let $k \in \{j \in N_l(x) \cap A_2 | j > i\}$. From the "only-if" part of the proof we know by (2), $i < k$, $x_i^N > 0$, and $x_k^N < d_k$ it holds that $r_i'(x_i^N) \geq r_k'(x_k^N)$. Combining this with (4.6) implies

$$r_i'(x_i) > r_i'(x_i^N) \geq r_k'(x_k^N) \geq r_k'(x_k).$$

Hence, $r_i'(x_i) > r_k'(x_k)$ which violates (1) by $i, k \in N_l(x), x_i < d_i$, and $x_k > 0$.

Secondly, assume $\{j \in N_l(x) \cap A_2 | j > i\} = \emptyset$[3]. Note that

$$\sum_{j=1}^{\max(N_l(x))} x_j = \sum_{j=1}^{\max(N_l(x))} e_j \geq \sum_{j=1}^{\max(N_l(x))} x_j^N.$$

Here, the equality follows from definition of $N_l(x)$, the inequality from feasibility of x^N. Since $x_i^N > x_i$, there exists at least one agent $k \leq \max(N_l(x))$

[3] For simplicity, for a set of agents $A \subset N$, we write $\max(A)$ rather than $\max\{i \in N | i \in A\}$ and $\min(A)$ rather than $\min\{i \in N | i \in A\}$, in order to describe, respectively, the most downstream and most upstream agent in A.

such that $x_k^N < x_k$ (i.e. $k \in A_2$). Let $k = \max\{j \in A_2 | j \leq \max(N_l(x))\}$, i.e. k is the most downstream agent in A_2, which is upstream of $\max(N_l(x))$. From the fact that $\{j \in N_l(x) \cap A_2 | j > i\} = \emptyset$ it follows that $k < i$.
Now we show that $k \in N_{\hat{l}}(x^N)$. Suppose $k \notin N_{\hat{l}}(x^N)$. By the fact that $k < i, i \in N_{\hat{l}}(x^N), k \notin N_{\hat{l}}(x^N)$, and $k = \max\{j \in A_2 | j \leq \max(N_l(x))\}$ it follows that there is no agent $j \in A_2$ (i.e. $x_j > x_j^N$) such that $\min(N_{\hat{l}}(x^N)) \leq j \leq \max(N_l(x))$. By $x_i^N > x_i$ it holds that

$$\sum_{j=\min(N_{\hat{l}}(x^N))}^{\max(N_l(x))} x_j < \sum_{j=\min(N_{\hat{l}}(x^N))}^{\max(N_l(x))} x_j^N. \tag{4.7}$$

Now the contradiction is established by

$$\begin{aligned}
\sum_{j=1}^{\max(N_l(x))} e_j &= \sum_{j=1}^{\max(N_l(x))} x_j \\
&= \sum_{j=1}^{\min(N_{\hat{l}}(x^N))-1} x_j + \sum_{j=\min(N_{\hat{l}}(x^N))}^{\max(N_l(x))} x_j \\
&< \sum_{j=1}^{\min(N_{\hat{l}}(x^N))-1} x_j + \sum_{j=\min(N_{\hat{l}}(x^N))}^{\max(N_l(x))} x_j^N \\
&\leq \sum_{j=1}^{\min(N_{\hat{l}}(x^N))-1} e_j + \sum_{j=\min(N_{\hat{l}}(x^N))}^{\max(N_l(x))} x_j^N \\
&= \sum_{j=1}^{\min(N_{\hat{l}}(x^N))-1} x_j^N + \sum_{j=\min(N_{\hat{l}}(x^N))}^{\max(N_l(x))} x_j^N \\
&= \sum_{j=1}^{\max(N_l(x))} x_j^N.
\end{aligned}$$

The first and third equality holds by definition, the first inequality by (4.7), and the second inequality by feasibility of x. Hence, $k \in N_{\hat{l}}(x^N)$. From the "only-if" part of the proof we know by (1) that since $x_i^N > 0$ and $x_k^N < d_k$ it holds that $r_i'(x_i^N) \geq r_k'(x_k^N)$. Combining this with $r_i'(x_i^N) < r_i'(x_i)$ and (4.6) tells us that

$$r_i'(x_i) > r_i'(x_i^N) \geq r_k'(x_k^N) \geq r_k'(x_k).$$

Hence, $r_i'(x_i) > r_k'(x_k)$. Since $k < i, x_k > 0$, and $x_i < d_i$ this violates (2).

A similar analysis can be made if for all $i \in A_1, r_i'(x_i^N) = r_i'(x_i)$, but there exists an agent $k \in A_2$ such that $r_k'(x_k^N) > r_k'(x_k)$. Then again, either (1) or (2) is violated. □

Example 4.2.1 shows how Theorem 4.2.1 can be used to check the optimality of an assignment x.

Example 4.2.1. Let $(N, e, d, r) \in RSA^N$ with $N = \{1, \ldots, 8\}$ and assignment $x \in \mathbb{R}^N$ as described in Figure 4.2 and Table 4.1. In Figure 4.2 the structure of the river is outlined by means of a direct graph. Here, a node refers to an agent i and its demand d_i (denoted by $i : d_i$), the incoming arc refers to the inflow e_i on the territory of the agent, the outgoing arc denotes the assigned water x_i, and the arcs in-between agents outlines the transfer from the upstream to the downstream agent (i.e. $\sum_{j=1}^{i-1}(e_j - x_j)$). For example, agent 5 has an inflow of $e_5 = 2$, demand $d_5 = 3$ and is assigned $x_5 = 1$. The total inflow upstream of agent 5 equals 12, of which 2 is assigned to agents in $\{1, \ldots, 4\}$. Hereby, 10 units of water is transferred to agent 5. The reward functions for all agents are outlined in Table 4.1.

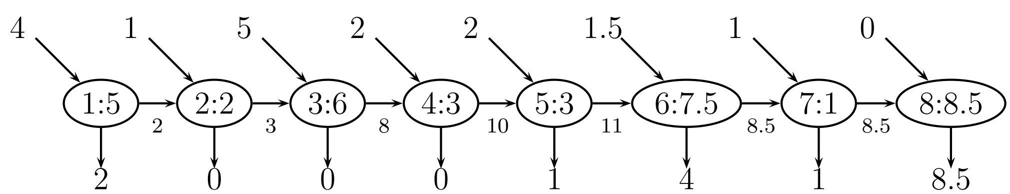

Figure 4.2: The inflows, demands and assignment x for the RSA-problem of Example 4.2.1

Feasibility can be verified from Figure 4.2 as follows. First, efficiency holds by the fact that agent 8 is assigned his inflow (of 0) and transfer from upstream agents (of 8.5). Secondly, since all in-between arc are non-negative, the sum of assigned water does not exceeds the available water. At last, each agent is assigned a non-negative amount of water which does not exceed its demand.

Now we use the conditions (1) and (2) of Theorem 4.2.1 in order to check optimality of the assignment x. Clearly, $P(x) = \{8\}, m(x) = 1$, and $N_1(x) = N$. This can be observed from Figure 4.2 by the fact that there is

Agent	$r_i(z)$	x_i	$r'_i(x_i)$
1	$12z - z^2$	2	8
2	$8z - z^2$	0	8
3	$2z - \frac{1}{6}z^2$	0	2
4	$4z - \frac{2}{3}z^2$	0	4
5	$12z - 2z^2$	1	8
6	$12z - \frac{1}{2}z^2$	4	8
7	$13z - z^2$	1	11
8	$10z$	8.5	10

Table 4.1: The reward functions $r(z)$, and the derivative $r'(x)$ for assignment x for the 8-person RSA problem Example 4.2.1

no arc in-between agents with 0 transfer. Now, condition (1) implies (2) such that, in order to check optimality of x, it suffices to check condition (1) for all pairs of agents in N. For this, we check for $r'_i(x_i) \geq r'_k(x_k)$ where $i, k \in N$ is such that

1. $x_i = d_i$ and $x_k \in [0, d_k)$, or

2. $x_i \in (0, d_i]$ and $x_k = 0$, or

3. $x_i \in (0, d_i)$ and $x_k \in (0, d_k)$.

1. Note that $\{i \in N | x_i = d_i\} = \{7, 8\}$ where $r'_7(x_7) = 11$ and $r'_8(x_8) = 10$. Since $r'_k(x_k) \leq 8$ for all $k \in N \setminus \{7, 8\}$, it is readily verified that condition (1) holds.

2. Now $\{k \in N | x_k = 0\} = \{2, 3, 4\}$ with $r'_2(x_2) = 8, r'_3(x_3) = 2$, and $r'_4(x_4) = 4$. Condition (1) follows from the fact that $r'_i(x_i) \geq 8$ for all $i \in N \setminus \{2, 3, 4\}$.

3. At last, $\{j \in N | x_j \in (0, d_j)\} = \{1, 5, 6\}$, with $r'_1(x_1) = r'_5(x_5) = r'_6(x_6) = 8$ which satisfies condition (1) such that $x \in X(N, e, d, r)$. $\diamondsuit$

Corollary 4.2.2 follows from the fact that for each $l \in \{1, \ldots, m(x)\}$, the most downstream agent in $N_l(x)$, extracts all water such that there is no flow in-between agents in different sets of an x-partition:

Corollary 4.2.2. *Let* $(N, e, d, r) \in RSA^N$, $x \in X(N, e, d, r)$ *and* $\{N_1(x), \ldots, N_{m(x)}(x)\}$ *be the corresponding* x*-partition. Then*[4]

$$v(N, e, d, r) = \sum_{l=1}^{m(x)} v(N_l(x), e|_{N_l(x)}, d|_{N_l(x)}, r|_{N_l(x)}).$$

Theorem 4.2.1 characterizes optimal assignments. We now present a methodology to systematically determine an optimal assignment. We first discuss the general structure of this methodology, and then this method is illustrated in Example 4.2.2.

Methodology 4.2.3.
Input: $(N, e, d, r) \in RSA^N$
Output: $x \in X(N, e, d, r)$

Step 1. Initialize $x_j = e_j$ for all $j \in N$, and $i = 2$.

Step 2. If $x_i = d_i$ or $\{j \in N | j < i, x_j > 0, r'_j(x_j) < r'_i(x_i)\} = \emptyset$, then go to *Step 7*.
If $x_i < d_i$ and $\{j \in N | j < i, x_j > 0, r'_j(x_j) < r'_i(x_i)\} \neq \emptyset$, then go to *Step 3*.

Step 3. Define $H = \{j \in N | j < i, x_j > 0, r'_j(x_j) < r'_i(x_i)\}$

Step 4. Define $K = \{j \in H | r'_j(x_j) = \min\{r'_h(x_h) | h \in H\}\}$

Step 5. Define $\epsilon \in \mathbb{R}^K$ by solving

$$\begin{aligned} &\max \sum_{k \in K} \epsilon_k \\ s.t.\ & r'_k(x_k - \epsilon_k) = r'_j(x_j - \epsilon_j) && \text{for all } k, j \in K && (4.8) \\ & r'_k(x_k - \epsilon_k) \leq \min\{r'_j(x_j) | j \in H \setminus K\} && \text{for all } k \in K && (4.9) \\ & r'_k(x_k - \epsilon_k) \leq r'_i(x_i + \sum_{k \in K} \epsilon_k) && \text{for all } k \in K && (4.10) \end{aligned}$$

Step 6. Update x by $x_k := x_k - \epsilon_k$ for all $k \in K$, and $x_i := x_i + \sum_{k \in K} \epsilon_k$. Next, return to *Step 2*.

[4] $e|_S \in \mathbb{R}^S$ denotes the restricted vector of inflows for agents in $S \subset N$ with respect to $e \in \mathbb{R}^N$; $r|_S$ refers to $\{r_j(z)\}_{j \in S}$

Step 7. If $i = |N|$, then stop.
If $i < |N|$, then let $i := i + 1$. Next, return to *Step 2.*

In each step of Methodology 4.2.3, an optimal assignment is determined for the set of upstream agents up to agent i. However, for some agents, *Step 2 - Step 6* are repeated a finite number of times. This is illustrated in the Example 4.2.2. Although there always is an $\{\epsilon_k\}_{k\in K}$ such that conditions (4.8)-(4.10) are satisfied, the complexity of maximizing $\sum_{k\in K}\epsilon_k$ depends on the complexity of the reward functions. For Example 4.2.2 we use SOLVER on MS-Excel, in order to obtain $\{\epsilon_k\}_{k\in K}$.

Example 4.2.2. Reconsider Example 4.2.1 with $N = \{1, \ldots, 8\}$ as described in Figure 4.2 and Table 4.1. The assignment of water is not yet determined.

Methodology 4.2.3 sequentially determines an optimal assignment for agents in $\{1, \ldots, i\}$ for increasing i. Table 4.2 outlines these assignments $x^i \in \mathbb{R}^N$. Here $(x^i_j)_{j\leq i}$ refers to the optimal assignment for agents in $\{1, \ldots, i\}$ obtained by Methodology 4.2.3, i.e. $(x^i_j)_{j\leq i} \in X(\{1, \ldots, i\}, e|_{\{1,\ldots,i\}}, d|_{\{1,\ldots,i\}}, r|_{\{1,\ldots,i\}})$ (for simplicity written as $X(\{1, \ldots, i\})$), and $x^i_j = e_j$ for all $j > i$.

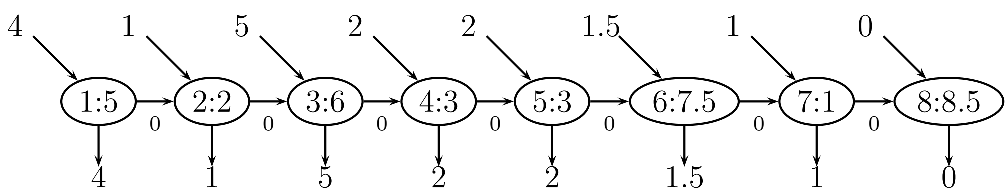

Figure 4.3: Initial assignment $x = e$ for the RSA-problem in Example 4.2.2

Step 1. Initially, all agents obtain their inflow, such that $x = e = (4, 1, 5, 2, 2, 1.5, 1, 0)$ and no water is transferred. This assignment is outlined in Figure 4.3. Let $i = 2$.

Step 2 (i=2). Now $x_2 = 1 < d_2$ such that agent 2 is not yet satisfied. Furthermore, $\{j \in N | j < 2, x_j > 0, r'_j(x_j) < r'_2(x_2)\} = \{1\}$, which yields that agent 1 is assigned a positive amount of water and the marginal contribution of agent 1 (derivative of the reward function of agent 1 in the current assignment) is lower than the marginal contribution of agent 2. Hence, there exists a feasible transfer from agent 1 to agent 2, which increases the total joint reward. In order to determine the amount of water transfer, we continue considering *Step 3.*

Step 3 (i=2). Define $H = \{j \in N | j < 2, x_j > 0, r'_j(x_j) < r'_2(x_2)\} = \{1\}$.

Step 4 (i=2). Let K be the subset of H containing all agents with the lowest derivative of all agents in H. Clearly, $H = K$.

Step 5 (i=2). Now determine the amount of water transfer by solving the following maximization problem.

$$\begin{aligned} \max \quad & \epsilon_1 \\ s.t. \quad & r'_1(x_1 - \epsilon_1) \leq r'_2(x_2 + \epsilon_1) \end{aligned}$$

Recall that by $|H| = 1$, equality (4.8) and inequality (4.9) do not apply. For $\epsilon_1 = 0.5$ it holds that $r'_1(x_1 - \epsilon_1) = r'_1(3.5) = r'_2(x_2 + \epsilon_1) = r'_2(1.5) = 5$. Hence, transferring 0.5 from agent 1 to agent 2 leads to an assignment in which the marginal contribution of agent 1 and 2 coincide.

Step 6 (i=2). Let x be such that $x_1 := x_1 - \epsilon_1 = 3.5$ and $x_2 := x_2 + \epsilon_1 = 1.5$. Next, return to *Step 2.*

Step 2 (i=2). Now, $\{j \in N | j < 2, x_j > 0, r'_j(x_j) < r'_2(x_2)\} = \emptyset$. Transferring more water (if feasible) from agent 1 to agent 2 will not increase the total joint reward and therefore we go to *Step 7.* Since this finishes the optimization for set of agents $\{1, 2\}$, assignment $(x_1^2, x_2^2) \in X(\{1, 2\})$ is displayed in Table 4.2. Optimality is checked by Theorem 4.2.1.

Step 7 (i=2). Let $i = 3$.

Step 2 (i=3). By $\{j \in N | j < 3, x_j > 0, r'_j(x_j) < r'_3(x_3)\} = \emptyset$, it holds that there does not exists an agent upstream of agent 3 with a marginal contribution for the current assignment lower than the marginal contribution of agent 3. Hereby, no water is transferred to agent 3. Consequently, we go to *Step 7*, $x^2 = x^3$ and $\{x_j^3\}_{j \leq 3} \in X(\{1, 2, 3\})$ and can be found in Table 4.2.

Step 7 (i=3). Let $i = 4$.

Step 2 (i=4). By $x_4 = 2 < d_4 = 3$ and $\{j \in N | j < 4, x_j > 0, r'_j(x_j) < r'_4(x_4)\} = \{3\}$, we consider transferring water from agent 3 to agent 4.

Step 3-4 (i=4). Define $H = K = \{3\}$.

Step 5 (i=4). Solving the following maximization problem determines the transfer from agent 3 to 4.

$$\begin{aligned} \max \quad & \epsilon_3 \\ s.t. \quad & r_3'(x_3 - \epsilon_3) \leq r_4'(x_4 + \epsilon_3) \end{aligned}$$

For $\epsilon_3 = 0.6$ it holds that $r_3'(4.4) = r_4'(2.6) = 0.533$.

Step 6 (i=4). Let x be such that $\{x_j\}_{j \leq 4} = (3.5, 1.5, 4.4, 2.6)$. Next, return to *Step 2*.

Step 2 (i=4). From $\{j \in N | j < 4, x_j > 0, r_j'(x_j) < r_4'(x_4)\} = \emptyset$, it follows that there does not exist a feasible transfer to agent 4 which increases the total joint reward. Therefore, we go to *Step 7* and $\{x_j^4\}_{j \leq 4} \in X(\{1, \ldots, 4\})$ can be found in Table 4.2.

Step 7 (i=4). Let $i = 5$.

Step 2 (i=5). By $x_5 < d_5$ and $\{j \in N | j < 5, x^j > 0, r_j'(x_j) < r_5'(x_5)\} = \{3, 4\}$, it holds that agent 5 is not yet satisfied, and agent 3 and 4 both are currently assigned a positive amount of water and have a marginal contribution lower than agent 5. Therefore, we consider transferring water from these agents to agent 5.

Step 3 (i=5). Define $H = \{3, 4\}$.

Step 4 (i=5). Since, $r_3'(x_3) = r_4'(x_4)$, it holds that $H = K = \{3, 4\}$. Now agents 3 and 4 simultaneously transfer water to agent 5, such that their derivatives remain mutually equal (by inequality (4.8)).

Step 5-6 (i=5). Determine $\epsilon \in \mathbb{R}^K$ where

$$\begin{aligned} \max \quad & \epsilon_3 + \epsilon_4 \\ s.t. \quad & r_3'(x_3 - \epsilon_3) = r_4'(x_4 - \epsilon_4) \\ & r_3'(x_3 - \epsilon_3) \leq r_5'(x_5 + \epsilon_3 + \epsilon_4) \\ & r_4'(x_4 - \epsilon_4) \leq r_5'(x_5 + \epsilon_3 + \epsilon_4) \end{aligned}$$

For $\epsilon_3 = \frac{13}{20}$ and $\epsilon_4 = \frac{13}{80}$ it holds that $r_3'(3\frac{3}{4}) = r_4'(2\frac{7}{16}) = r_5'(2\frac{13}{16}) = 0.75$. Now let x such that $\{x_j\}_{j \leq 5} = (3.5, 1.5, 3\frac{3}{4}, 2\frac{7}{16}, 2\frac{13}{16})$ and return to *Step 2*.

Step 2 (i=5). By $\{j \in N | j < 5, x_j > 0, r_j'(x_j) < r_5'(x_5)\} = \emptyset$, we go to *Step 7*. Now, $\{x_j^5\}_{j \leq 5} \in X(\{1, \ldots, 5\})$ can be found in Table 4.2.

Step 7 (i=5). Let $i = 6$.

Step 2 (i=6). Now, $x_6 = 1.5 < d_6 = 7.5$ and $\{j \in N | j < 6, x_j > 0, r'_j(x_j) < r'_6(x_6)\} = \{1, \ldots, 5\}$. Consequently, we consider *Step 3*.

Step 3 (i=6). Define $H = \{1, \ldots, 5\}$.

Step 4 (i=6). Let K be the subset of H containing all agents with the lowest derivative of all agents in H. Since $r'_1(x_1) = r'_2(x_2) = 5 > r'_3(x_3) = r'_4(x_4) = r'_5(x_5) = 0.75$, it holds that $K = \{3, 4, 5\}$. Hence, although agents 1 and 2 both have a marginal contribution lower than agent 6, we first only consider the transfer of agents 3, 4 and 5, since their derivatives are even lower. These agents simultaneously transfer water to agent 6.

Step 5-6 (i=6). Now

$$\begin{aligned}
\max \quad & \epsilon_3 + \epsilon_4 + \epsilon_5 \\
s.t. \quad & r'_3(x_3 - \epsilon_3) = r'_4(x_4 - \epsilon_4) = r'_5(x_5 - \epsilon_5) \\
& r'_3(x_3 - \epsilon_3) \leq \min\{r'_j(x_j) | j \in \{1, 2\}\} = 5 \\
& r'_4(x_4 - \epsilon_4) \leq 5 \\
& r'_5(x_5 - \epsilon_5) \leq 5 \\
& r'_3(x_3 - \epsilon_3) \leq r'_6(x_6 + \epsilon_3 + \epsilon_4 + \epsilon_5) \\
& r'_4(x_4 - \epsilon_4) \leq r'_6(x_6 + \epsilon_3 + \epsilon_4 + \epsilon_5) \\
& r'_5(x_5 - \epsilon_5) \leq r'_6(x_6 + \epsilon_3 + \epsilon_4 + \epsilon_5)
\end{aligned}$$

For $\epsilon_3 = 3\frac{3}{4}, \epsilon_4 = \frac{15}{16}$, and $\epsilon_5 = \frac{5}{16}$ it holds that $r'_3(0) = r'_4(1.5) = r'_5(2.5) = 2 < r'_6(6.5) = 5.5$. Note that agent 3 runs out of water, such that transferring more water to agent 6 violates equality (4.8). Now let x such that $\{x_j\}_{j \leq 6} = (3.5, 1.5, 0, 1.5, 2.5, 6.5)$[5] and return to *Step 2*.

Step 2 (i=6). Again $x_6 < d_6$ and $\{j \in N | j < 6, x_j > 0, r'_j(x_j) < r'_6(x_6)\} \neq \emptyset$. Since agent 6 is not yet satisfied and there still are upstream agents with a positive amount of assigned water and a marginal contribution lower than agent 6, we again consider transferring water to agent 6.

Step 3-5 (i=6). Define $H = \{1, 2, 4, 5\}$ and $K = \{4, 5\}$. Hence, this involves another set of upstream agents where agent 3 is deleted from previous

[5]Note that x is an intermediate assignment to go from x^5 to x^6 and is not optimal for $\{1, \ldots, 5\}$

set K by $x_3 = 0$. Now

$$\begin{aligned}
\max \quad & \epsilon_4 + \epsilon_5 \\
s.t. \quad & r'_4(x_4 - \epsilon_4) = r'_5(x_5 - \epsilon_5) \\
& r'_4(x_4 - \epsilon_4) \leq 5 \\
& r'_5(x_5 - \epsilon_5) \leq 5 \\
& r'_4(x_4 - \epsilon_4) \leq r'_6(x_6 + \epsilon_4 + \epsilon_5) \\
& r'_5(x_5 - \epsilon_5) \leq r'_6(x_6 + \epsilon_4 + \epsilon_5)
\end{aligned}$$

For $\epsilon_4 = 0.75$, and $\epsilon_5 = 0.25$ it holds that $r'_4(0.75) = r'_5(2.25) = 3 < r'_6(7.5) = 4.5$. Now, inequality (4.10) in binding since assigning more than 7.5 to agent 6 is infeasible by demand d_6.

Step 6 (i=6). Let x such that $\{x_j\}_{j \leq 6} = (3.5, 1.5, 0, 0.75, 2.25, 7.5)$ and return to *Step 2*.

Step 2 (i=6). Now $x_6 = 7.5 = d_6$ tells us that agent 6 is totally satisfied. Hereby, transferring more water to this agent is infeasible. Hence, we go to *Step 7* and assignment, $\{x^6_j\}_{j \leq 6} \in X(\{1, \ldots, 6\})$ is found in Table 4.2.

Step 7 (i=6). Let $i = 7$.

Step 2 (i=7). By $x_7 = d_7$ it holds that agent 7 is already satisfied and there is no feasible transfer of water to agent 7. Hereby, we go to *Step 7.* As displayed in Table 4.2, $x^6 = x^7$ and $\{x^7_j\}_{j \leq 7} \in X(\{1, \ldots, 7\})$.

Step 7 (i=7). Let $i = 8$.

Step 2 (i=8). Now $x_8 < d_8$ and $\{j \in N | j < 8, x_j > 0, r'_j(x_j) < r'_8(x_8)\} = \{1, 2, 4, 5, 6\}$. Consequently, we go to *Step 3.*

Step 3-6 (i=8). Define $H = \{1, 2, 4, 5, 6\}$ and $K = \{4, 5\}$. Now

$$\begin{aligned}
\max \quad & \epsilon_4 + \epsilon_5 \\
s.t. \quad & r'_4(x_4 - \epsilon_4) = r'_5(x_5 - \epsilon_5) \\
& r'_4(x_4 - \epsilon_4) \leq \min\{r'_j(x_j) | j \in \{1, 2, 6\}\} = 4.5 \\
& r'_5(x_5 - \epsilon_5) \leq 4.5 \\
& r'_4(x_4 - \epsilon_4) \leq r'_8(x_8 + \epsilon_4 + \epsilon_5) \\
& r'_5(x_5 - \epsilon_5) \leq r'_8(x_8 + \epsilon_4 + \epsilon_5)
\end{aligned}$$

For $\epsilon_4 = 0.75$, and $\epsilon_5 = 0.25$ it holds that $r'_4(0) = r'_5(2) = 4 < r'_8(1) = 10$. Note that agent 4 runs out of water. Let $x := (3.5, 1.5, 0, 0, 2, 7.5, 1, 1)$ and return to *Step 2*.

Step 2 (i=8). Now $x_8 < d_8$ and $\{j \in N | j < 8, x_j > 0, r'_j(x_j) < r'_8(x_8)\} = \{1, 2, 5, 6\}$. Hence, we again consider *Step 3*.

Step 3-6 (i=8). Now, transferring water to agent 8 is considered for $H = \{1, 2, 5, 6\}$ and $K = \{5\}$. Then,

$$\begin{aligned} \max \quad & \epsilon_5 \\ s.t. \quad & r'_5(x_5 - \epsilon_5) \leq \min\{r'_j(x_j) | j \in \{1, 2, 6\}\} = 4.5 \\ & r'_5(x_5 - \epsilon_5) \leq r'_8(x_8 + \epsilon_5) \end{aligned}$$

For $\epsilon_5 = \frac{1}{8}$ it holds that $r'_5(1\frac{7}{8}) = 4.5$. Note that inequality (4.9) is binding. Hence, in set $H \setminus K = \{1, 2, 6\}$ there are agents whose derivative is now equal to the derivative of agent 5. Now, let $x := (3.5, 1.5, 0, 0, 1\frac{7}{8}, 7.5, 1, 1\frac{1}{8})$ and return to *Step 2*.

Step 2 (i=8). Now, $x_8 < d_8$ and $\{j \in N | j < 8, x_j > 0, r'_j(x_j) < r'_8(x_8)\} = \{1, 2, 5, 6\}$ such that again the transfer to agent 8 is considered.

Step 3-6 (i=8). Define $H = \{1, 2, 5, 6\}$ and $K = \{5, 6\}$. The derivatives of agent 5 and 6 coincides such that agent 6 is added to set K. Then,

$$\begin{aligned} \max \quad & \epsilon_5 + \epsilon_6 \\ s.t. \quad & r'_5(x_5 - \epsilon_5) = r'_6(x_6 - \epsilon_6) \\ & r'_5(x_5 - \epsilon_5) \leq \min\{r'_j(x_j) | j \in \{1, 2\}\} = 5 \\ & r'_6(x_6 - \epsilon_6) \leq 5 \\ & r'_5(x_5 - \epsilon_5) \leq r'_8(x_8 + \epsilon_5 + \epsilon_6) \\ & r'_6(x_6 - \epsilon_6) \leq r'_8(x_8 + \epsilon_5 + \epsilon_6) \end{aligned}$$

For $\epsilon_5 = \frac{1}{8}$, and $\epsilon_6 = 0.5$ it holds that $r'_5(1.75) = r'_6(7) = 5$. Now $x := (3.5, 1.5, 0, 0, 1.75, 7, 1, 1.75)$ and return to *Step 2*.

Step 2 (i=8). By $x_8 < d_8$ and $\{j \in N | j < 8, x_j > 0, r'_j(x_j) < r'_8(x_8)\} = \{1, 2, 5, 6\}$, it follows that we again should determine the set of agents for which transferring water to agent 8 is beneficial.

Step 3-6 (i=8). Define $H = K = \{1, 2, 5, 6\}$. Hence, agents 1 and 2 are added to set K. Then,

$$\begin{aligned}
\max \quad & \epsilon_1 + \epsilon_2 + \epsilon_5 + \epsilon_6 \\
s.t. \quad & r_1'(x_1 - \epsilon_1) = r_2'(x_2 - \epsilon_2) = r_5'(x_5 - \epsilon_5) = r_6'(x_6 - \epsilon_6) \\
& r_1'(x_1 - \epsilon_1) \leq r_8'(x_8 + \epsilon_1 + \epsilon_2 + \epsilon_5 + \epsilon_6) \\
& r_2'(x_2 - \epsilon_2) \leq r_8'(x_8 + \epsilon_1 + \epsilon_2 + \epsilon_5 + \epsilon_6) \\
& r_5'(x_5 - \epsilon_5) \leq r_8'(x_8 + \epsilon_1 + \epsilon_2 + \epsilon_5 + \epsilon_6) \\
& r_6'(x_6 - \epsilon_6) \leq r_8'(x_8 + \epsilon_1 + \epsilon_2 + \epsilon_5 + \epsilon_6)
\end{aligned}$$

For $\epsilon_1 = 1.5, \epsilon_2 = 1.5, \epsilon_5 = 0.75$, and $\epsilon_6 = 3$ it holds that $r_1'(2) = r_2'(0) = r_5'(1) = r_6'(4) = 8$ and $x_8 = d_8 = 8.5$. Let $x := (2, 0, 0, 0, 1, 4, 1, 8.5)$ and return to *Step 2*.

Step 2 (i=8). By $x_8 = d_8$, it follows that there is no further transfer possible to agent 8. Therefore, go to *Step 7* and $x^8 = (2, 0, 0, 0, 1, 4, 1, 8.5)$ is outlined in Table 4.2.

Step 7 (i=8). Now $i = 8 = |N|$, finishes the methodology of obtaining assignment $x = x^8$, outlined in Figure 4.2.

From Example 4.2.1 it follows that $x \in X(N, e, d, r)$.

Agent	x^1	x^2	x^3	x^4	x^5	x^6	x^7	x^8
1	4	$3\frac{1}{2}$	$3\frac{1}{2}$	$3\frac{1}{2}$	$3\frac{1}{2}$	$3\frac{1}{2}$	$3\frac{1}{2}$	2
2	1	$1\frac{1}{2}$	$1\frac{1}{2}$	$1\frac{1}{2}$	$1\frac{1}{2}$	$1\frac{1}{2}$	$1\frac{1}{2}$	0
3	5	5	5	$4\frac{2}{5}$	$3\frac{3}{4}$	0	0	0
4	2	2	2	$2\frac{3}{5}$	$2\frac{7}{16}$	$\frac{3}{4}$	$\frac{3}{4}$	0
5	2	2	2	2	$2\frac{13}{16}$	$2\frac{1}{4}$	$2\frac{1}{4}$	1
6	$1\frac{1}{2}$	$1\frac{1}{2}$	$1\frac{1}{2}$	$1\frac{1}{2}$	$1\frac{1}{2}$	$7\frac{1}{2}$	$7\frac{1}{2}$	4
7	1	1	1	1	1	1	1	1
8	0	0	0	0	0	0	0	$8\frac{1}{2}$

Table 4.2: Assignments x^i for Example 4.2.2 using Methodology 4.2.3

$\diamondsuit$

Consider assignments x^i and x^{i+1} for $i \in N \backslash \{n\}$ as in Example 4.2.2. Theorem 4.2.1 allows us to check that x^i is an optimal assignment for $\{1, \ldots, i\}$ and x^{i+1} for $\{1, \ldots, i+1\}$. Table 4.2 shows that for each agent $j < i$ the amount of assigned water x_j^i decreases when downstream agents are added, i.e. $x_j^i \geq x_j^{i+1}$.

4.3 River Sharing Allocation Games

Consider a coalition $S \subset N$ of agents choosing their amount of extracted water in order to maximize the total joint reward. They may transfer available water to downstream coalition agents with a greater marginal contribution, provided this is feasible. However, water to be transferred to downstream coalition agents via the territory of an intermediate, non-coalition agent, referred to as 'individualist', involves *leakage* because individualists are assumed to act individualistic by extracting the amount of water in order to maximize individual payoffs by using all available water up to their demand.

An assignment of water to agents in coalition S is referred to as $x^S \in \mathbb{R}^S$. Such assignment allows us to recursively determine the amount of water extracted by agents in $N \setminus S$. For this, let $x^S \in \mathbb{R}^S$. Define $x^{N\setminus S} \in \mathbb{R}^{N\setminus S}$ recursively such that

$$x_{\min(N\setminus S)}^{N\setminus S} = \min\left\{e_{\min(N\setminus S)} + \sum_{j=1}^{\min(N\setminus S)-1} (e_j - x_j^S), d_{\min(N\setminus S)}\right\},$$

and for all $i \in N \setminus S, j \neq \min(N \setminus S)$

$$x_i^{N\setminus S} = \min\left\{e_i + \sum_{j\in S:j<i} (e_j - x_j^S) + \sum_{j\in N\setminus S:j<i} (e_j - x_j^{N\setminus S}), d_i\right\}. \quad (4.11)$$

An assignment x^S is called feasible for S if

$$x_i^S \in [0, d_i] \text{ for all } i \in S; \quad (4.12)$$

$$x_i^S \leq \sum_{j=1}^{i} e_j - \sum_{j\in N\setminus S:j<i} x_j^{N\setminus S} - \sum_{j\in S:j<i} x_j^S \text{ for all } i \in S; \quad (4.13)$$

$$x_{\max(S)}^S = \sum_{j=1}^{\max(S)} e_j - \sum_{j\in N\setminus S:j<\max(S)} x_j^{N\setminus S} - \sum_{j\in S:j<\max(S)} x_j^S. \quad (4.14)$$

Inflow constraint (4.13) tells us that for each agent $i \in S$ the amount of assigned water is lower than or equal to the amount of available water. Efficiency constraint (4.14) states that the most downstream agent in S is assigned all its inflow.

Let $F^S(N,e,d,r)$ denote the set of feasible assignments x^S. The maximum total joint reward $v^S(N,e,d,r)$ for coalition S is determined by

$$v^S(N,e,d,r) = \max\left\{\sum_{j\in S} r_j(x_j)\,\middle|\, x \in F^S(N,e,d,r)\right\}.$$

The set $X^S(N,e,d,r)$ of optimal assignments is given by

$$X^S(N,e,d,r) = \left\{x^S \in F^S(N,e,d,r)\,\middle|\,\sum_{j\in S} r_j(x_j) = v^S(N,e,d,r)\right\}.$$

Now we introduce the class of *river sharing allocation* (RSA) *games.* A *transferable utility* (TU) *game* is an ordered pair (N,v) where N is the finite set of agents, and v the characteristic function on 2^N, the set of all coalitions of N. For the remainder we refer to N as the set of *players* instead of *agents.* The function v assigns to every coalition $S \in 2^N$ a real number $v(S)$ with $v(\emptyset) = 0$. Here, $v(S)$ is called the worth or value of the coalition S. The set of all TU-games with set of players N is denoted by TU^N. Where no confusion arises, we write v rather then (N,v).

Now consider an RSA-problem (N,e,d,r). In the RSA-game v, associated to (N,e,d,r), we define the worth of coalition S as

$$v(S) = v^S(N,e,d,r).$$

A coalition S is called *connected* if for all $i,j \in S$ and k such that $i < k < j$ it holds that $k \in S$. Due to the fact that players in $N \setminus S$ act individualistic and $d_i \geq e_i$ for all $i \in N$, players upstream of S do not transfer any water to S. Due to feasibility, the most downstream agent in S extracts all water, transferring nothing to downstream individualists. Consequently, we have the following corollary.

Corollary 4.3.1. *Let $(N,e,d,r) \in RSA^N$, let $v \in TU^N$ be the corresponding RSA-game, and let $S \subset N$ be such that S is connected. Then, $(S, e|_S, d|_S, r|_S) \in RSA^S$ and*

$$v(S) = v(S, e|_S, d|_S, r|_S).$$

Now suppose S is not connected. Then S consists of p components such that $S = \bigcup_{l=1}^{p} S_l$, and for all $l \in \{1, \ldots, p\}$, S_l is maximally connected and $\max(S_{l-1}) < \min(S_l)$ for all $l \in \{2, \ldots, p\}$. For the remainder, *component* refers to a maximally connected set of agents. Take an optimal assignment $x^S \in X^S(N, e, d, r)$ for S. We recursively determine the *transfer* $t_l(x^S)$ as the total amount of water transferred to component l from the components $1, \ldots, l-1$. Clearly,

$$t_1(x^S) = 0.$$

Further, it holds that for an optimal assignment, either S_{l-1} transfers an amount of water to S_l greater than the intermediate leakage, or no water at all. Therefore, for all $l \in \{2, \ldots, p\}$, $t_l(x^S)$ can be recursively determined by

$$t_l(x^S) = \max\left\{ t_{l-1}(x^S) + \sum_{j \in S_{l-1}} (e_j - x_j^S) - \sum_{j=\max(S_{l-1})+1}^{\min(S_l)-1} (d_j - e_j), 0 \right\}.$$

Note that for all players j such that $\max(S_{l-1}) < j < \min(S_l)$, it holds that $j \in N \setminus S$. Furthermore, $\sum_{j=\max(S_{l-1})+1}^{\min(S_l)-1} (d_j - e_j)$ is referred to as the leakage to individualists in-between S_{l-1} and S_l. A *stream component* of S refers to a collection of components of S which decide to transfer water and accept all leakage to intermediate individualists. Let partition

$$\{\tilde{S}_1(x^S), \ldots, \tilde{S}_{q(x^S)}(x^S)\}$$

be the collection of stream components with respect to x^S where components S_l and S_{l+1} are in the same stream component if $t_{l+1}(x^S) > 0$ and are in different stream components if $t_{l+1}(x^S) = 0$. Formally, let

$$L(x^S) = \{l \in \{1, \ldots, p\} | t_l(x^S) = 0\},$$

i.e. $L(x^S)$ refers to the components of S which do not receive any inflow of upstream components. Without loss of generality, let $L(x^S) = \{l_1, l_2, \ldots, l_{q(x^S)}\}$, with $l_1 < l_2 < \cdots < l_{q(x^S)}$. Observe that agents upstream of S_1 act individualistic such that $l_1 = 1$. Note that $q(x^S) \leq p$. Then, for all $i \in \{1, \ldots, q(x^S) - 1\}$, it holds that

$$\tilde{S}_i(x^S) = \bigcup_{k=l_i}^{l_{i+1}-1} S_k,$$

and

$$\tilde{S}_{q(x^S)}(x^S) = \bigcup_{k=l_{q(x^S)}}^{p} S_k.$$

Now, we clarify RSA-games by means of Example 4.3.1.

Example 4.3.1. Reconsider the RSA problem of Example 4.2.2. Let $S = \{1,2,4,5,6\}$. One readily checks that for component $S_1 = \{1,2\}, x^{S_1} = (3.5, 1.5)$ is an optimal assignment with total joint reward 39.5, while for component $S_2 = \{4,5,6\}, x^{S_2} = (0, 1.1, 4.4)$ is an optimal assignment with total joint reward 53.9. Furthermore, $x^S = (x^{S_1}, x^{S_2})$ is an optimal assignment for S with total joint reward 39.5+53.9=93.4. Now, transferring water from S_1 to S_2 is not beneficial, i.e. there is no leakage involved. Consequently, as outlined in Figure 4.4, $q(x^S) = 2, \tilde{S}_1(x^S) = S_1, \tilde{S}_2(x^S) = S_2$, and $t_2(x^S) = 0$. Now consider $T = S \cup \{8\}$. Then one readily verifies that

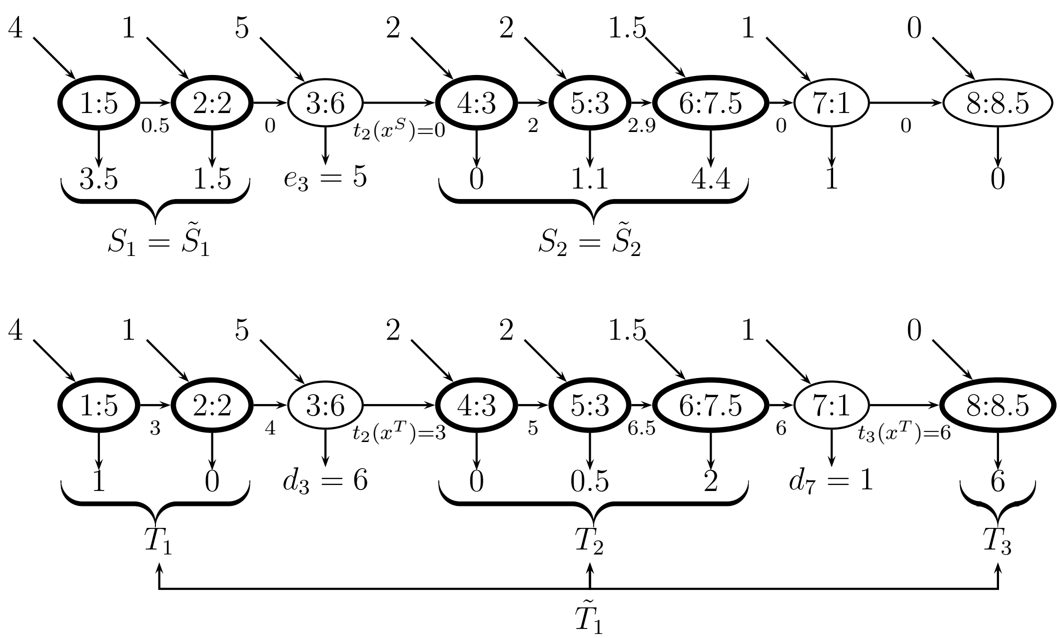

Figure 4.4: Assignments x^S and x^T for Example 4.3.1

$x^T = (1, 0, 0, 0.5, 2, 6)$ is an optimal assignment for T with total joint reward 116.5. This assignment is outlined in Figure 4.4. Hence, $q(x^T) = 1$ and T is one stream component. Now,

$$t_2(x^T) = (e_1 - x_1^T) + (e_2 - x_2^T) - (d_3 - e_3) = 3 + 1 - 1 = 3,$$

while

$$\begin{aligned} t_3(x^T) &= t_2(x^T) + (e_4 - x_4^T) + (e_5 - x_5^T) + (e_6 - x_6^T) - (d_7 - e_7) \\ &= 3 + 2 + 1.5 - 0.5 - 0 = 6. \end{aligned}$$

$\diamondsuit$

In Example 4.3.1, the optimal assignment to agents in S is such that no water is transferred from agents in component S_1 to agents in component S_2. However, by adding agent 8, this transfer needs to be reconsidered. This increases the general complexity of finding an optimal assignment for arbitrary coalitions. Although Methodology 4.2.3 can be used for the optimal assignment to components of S, the complexity arises from the fact that all possible combinations of neighboring components should be checked.

Let $\{\tilde{S}_1(x^S), \ldots, \tilde{S}_{q(x^S)}(x^S)\}$ be the collection of stream components of S with respect to the optimal assignment $x^S \in X^S(N, e, d, r)$. Since there is no transfer in-between stream components, it holds that

$$v(S) = \sum_{l=1}^{q(x^S)} v(\tilde{S}_l(x^S)). \tag{4.15}$$

This property is referred to as *stream component additivity*.

Lemma 4.3.2. *Let $(N, e, d, r) \in RSA^N$ and let $v \in TU^N$ be the corresponding RSA-game. Let $S \subset N$ with optimal assignment $x^S \in X^S(N, e, d, r)$ and let $T = \{i \in N | \min(S) \leq i \leq \max(S)\}$. If S is one stream component with respect to x^S, then*

$$v(T) \geq v(S) + \sum_{j \in T \setminus S} r_j(d_j).$$

Proof. Note that T is the connected set of all players in S and all its intermediate individualists. For x^S it holds that $S = \tilde{S}_1$ is a single stream component, such that all intermediate individualists $i \in T \setminus S$ extract their demand d_i. Define assignment $x \in \mathbb{R}^T$ and $x^{N \setminus T} \in \mathbb{R}^{N \setminus T}$ such that

$$x_i = \begin{cases} x_i^S, & \text{for all } i \in S, \\ x_i^{N \setminus S} = d_i, & \text{for all } i \in T \setminus S, \end{cases} \tag{4.16}$$

$$x_i^{N \setminus T} = x_i^{N \setminus S} = e_i \text{ for all } i \in N \setminus T. \tag{4.17}$$

In order to prove feasibility of x, we first prove that the constraints (4.12) - (4.14) hold for x.

First, (4.12) follows from $x_i = x_i^S \in [0, d_i]$ for all $i \in S$, and $x_i = d_i$ for all $i \in T \setminus S$. Secondly, observe that from the feasibility of x^S, i.e. from inequality (4.13), it follows that for all $i \in S$,

$$x_i = x_i^S \leq \sum_{j=1}^{i} e_j - \sum_{j \in N \setminus S : j < i} x_j^{N \setminus S} - \sum_{j \in S : j < i} x_j^S.$$

Furthermore, by equality (4.11) it holds for all $i \in T \setminus S$, that

$$x_i = d_i \leq \sum_{j=1}^{i} e_j - \sum_{j \in N \setminus S : j < i} x_j^{N \setminus S} - \sum_{j \in S : j < i} x_j^S.$$

Now, assignment x satisfies inequality (4.13) by the fact that for all $i \in T$,

$$\begin{aligned} x_i &\leq \sum_{j=1}^{i} e_j - \sum_{j \in N \setminus S : j < i} x_j^{N \setminus S} - \sum_{j \in S : j < i} x_j^S \\ &= \sum_{j=1}^{i} e_j - \sum_{j \in N \setminus T : j < i} x_j^{N \setminus S} - \sum_{j \in T \setminus S : j < i} x_j^{N \setminus S} - \sum_{j \in S : j < i} x_j^S \\ &= \sum_{j=1}^{i} e_j - \sum_{j \in N \setminus T : j < i} x_j^{N \setminus T} - \sum_{j \in T \setminus S : j < i} x_j - \sum_{j \in S : j < i} x_j \\ &= \sum_{j=1}^{i} e_j - \sum_{j \in N \setminus T : j < i} x_j^{N \setminus T} - \sum_{j \in T : j < i} x_j. \end{aligned}$$

The second equality holds by (4.16) and (4.17). Similarly, constraint (4.14) follows from

$$\begin{aligned}
x_{\max(T)} &= x_{\max(S)} \\
&= x^S_{\max(S)} \\
&= \sum_{j=1}^{\max(S)} e_j - \sum_{j\in N\setminus S: j<\max(S)} x_j^{N\setminus S} - \sum_{j\in S: j<\max(S)} x_j^S \\
&= \sum_{j=1}^{\max(S)} e_j - \sum_{j\in N\setminus T: j<\max(S)} x_j^{N\setminus S} - \sum_{j\in T\setminus S: j<\max(S)} x_j^{N\setminus S} \\
&\quad - \sum_{j\in S: j<\max(S)} x_j^S \\
&= \sum_{j=1}^{\max(S)} e_j - \sum_{j\in N\setminus T: j<\max(S)} x_j^{N\setminus T} - \sum_{j\in T\setminus S: j<\max(S)} x_j \\
&\quad - \sum_{j\in S: j<\max(S)} x_j \\
&= \sum_{j=1}^{\max(S)} e_j - \sum_{j\in N\setminus T: j<\max(S)} x_j^{N\setminus T} - \sum_{j\in T: j<\max(S)} x_j \\
&= \sum_{j=1}^{\max(T)} e_j - \sum_{j\in N\setminus T: j<\max(T)} x_j^{N\setminus T} - \sum_{j\in T: j<\max(T)} x_j .
\end{aligned}$$

The second and fifth equality holds by (4.16) and (4.17). The third equality by feasibility of x^S, i.e. equality (4.14). The first and last equality follows from $\max(T) = \max(S)$. Next, this implies,

$$\begin{aligned}
v(T) &= \max\left\{ \sum_{j\in T} r_j(x^T) \middle| x^T \in F^T(N,e,d,r) \right\} \\
&\geq \sum_{j\in T} r_j(x_j) \\
&= \sum_{j\in S} r_j(x_j^S) + \sum_{j\in T\setminus S} r_j(d_j) \\
&= v(S) + \sum_{j\in T\setminus S} r_j(d_j)
\end{aligned}$$

where the inequality holds by feasibility of x. $\square$

The following lemma states that two connected sets together can obtain a greater total joint reward, than both sets separately.

Lemma 4.3.3. *Let $(N, e, d, r) \in RSA^N$ and let $v \in TU^N$ be the corresponding RSA-game. Let $S, T \in 2^N$ be such that S is connected, T is connected, and $S \cap T = \emptyset$. Then,*

$$v(S) + v(T) \leq v(S \cup T).$$

Proof. Let $x^S \in X^S(N, e, d, r)$ and $x^T \in X^T(N, e, d, r)$. Define $x \in \mathbb{R}^{S \cup T}$ as $x_i = x_i^S$ for all $i \in S$ and $x_i = x_i^T$ for all $i \in T$.

Clearly, $x \in F^{S \cup T}(N, e, d, r)$. Hence, using Corollary 4.3.1

$$\begin{aligned} v(S) + v(T) &= \sum_{j \in S} r_j(x_j^S) + \sum_{j \in T} r_j(x_i^T) \\ &= \sum_{j \in S \cup T} r_j(x_j) \\ &\leq \max\left\{ \sum_{j \in S \cup T} r_j(x_j^{S \cup T}) \middle| x^{S \cup T} \in F^{S \cup T}(N, e, d, r) \right\} \\ &= v(S \cup T). \end{aligned}$$

□

The downstream incremental solution[6] $DI(v)$ is such that each player obtains the extra total joint reward he adds to the set of upstream players. Hence, for all $i \in N$,

$$DI_i(v) = v(\{1, \ldots, i\}) - v(\{1, \ldots, i-1\}).$$

Lemma 4.3.4 states that for each player $i \in N$ the downstream incremental solutions allocates an amount of rewards lower than or equal to the amount he obtains when demand d_i is assigned.

Lemma 4.3.4. *Let $(N, e, d, r) \in RSA^N$ and let $v \in TU^N$ be the corresponding RSA-game. Then, for all $i \in N$,*

$$r_i(d_i) \geq DI_i(v).$$

[6]The downstream incremental solution coincides with the marginal vector $m^\sigma(v)$ of the RSA-game v corresponding to the order $\sigma = \{1, \ldots, n\}$.

Proof. For $i = 1$, it holds that $DI_1 = r_1(e_1)$. Lemma 4.3.4 holds by $e_1 \leq d_1$.

Let $i \in N \setminus \{1\}, S = \{1, \ldots, i-1\}$, and $T = \{1, \ldots, i\}$. Let $x^S \in X^S(N, e, d, r)$ and $x^T \in X^T(N, e, d, r)$ be such that for all $j \in S$ it holds that $x^S_j \geq x^T_j$. From Methodology 4.2.3 it follows that such assignments exist. Now,

$$\begin{aligned} DI_i(v) &= v(\{1, \ldots, i\}) - v(\{1, \ldots, i-1\}) \\ &= \sum_{j=1}^{i} r_j(x^T_j) - \sum_{j=1}^{i-1} r_j(x^S_j) \\ &= r_i(x^T_i) + \sum_{j=1}^{i-1} \left(r_j(x^T_j) - r_j(x^S_j)\right) \\ &\leq r_i(x^T_i) \\ &\leq r_i(d_i). \end{aligned}$$

The first inequality follows from $x^S_j \geq x^T_j$ and r increasing, the last inequality from feasibility. $\square$

The core of a game consists of those efficient allocations such that no coalition has an incentive to split off from the grand coalition. Hence, the core is set of stable and efficient allocations, i.e.

$$C(v) = \left\{ y \in \mathbb{R}^N \middle| \sum_{j \in N} y_j = v(N), \sum_{j \in S} y_j \geq v(S) \text{ for all } S \in 2^N \right\}.$$

Theorem 4.3.5. *Let $(N, e, d, r) \in RSA^N$ and let $v \in TU^N$ be the corresponding RSA-game. Then $DI(v) \in C(v)$.*

Proof. Let $x^S \in X^S(N, e, d, r)$ and $\{\tilde{S}_1(x^S), \ldots, \tilde{S}_{q(x^S)}(x^S)\}$ be the collection of stream components of S with respect to x^S. Let $DI(v)$ be the corresponding downstream incremental solution. First note that $\sum_{j \in N} DI_j(v) = v(N)$ by definition. Secondly, we prove that $\sum_{j \in S} DI_j(v) \geq v(S)$. For all $l \in \{1, \ldots, q(x^S)\}$ let $T_l = \{i \in N | \min(\tilde{S}_l(x^S)) \leq i \leq \max(\tilde{S}_l(x^S))\}$ (i.e. the set of all players in $\tilde{S}_l(x^S)$ and all its intermediate individualists), and $U_l = \{i \in N | i < \min(\tilde{S}_l(x^S))\}$ (i.e. the set of all upstream players of $\tilde{S}_l(x^S)$).

Then

$$
\begin{aligned}
v(S) &= \sum_{l=1}^{q(x^S)} v(\tilde{S}_l(x^S)) \\
&\leq \sum_{l=1}^{q(x^S)} \left(v(T_l) - \sum_{j \in T_l \setminus \tilde{S}_l(x^S)} r_j(d_j) \right) \\
&\leq \sum_{l=1}^{q(x^S)} \left(v(T_l \cup U_l) - v(U_l) - \sum_{j \in T_l \setminus \tilde{S}_l(x^S)} r_j(d_j) \right) \\
&= \sum_{l=1}^{q(x^S)} \left(\sum_{j \in T_l} DI_j(v) - \sum_{j \in T_l \setminus \tilde{S}_l(x^S)} r_j(d_j) \right) \\
&= \sum_{l=1}^{q(x^S)} \left(\sum_{j \in \tilde{S}_l(x^S)} DI_j(v) + \sum_{j \in T_l \setminus \tilde{S}_l(x^S)} (DI_j(v) - r_j(d_j)) \right) \\
&\leq \sum_{l=1}^{q(x^S)} \sum_{j \in \tilde{S}_l(x^S)} DI_j(v) \\
&= \sum_{j \in S} DI_j(v).
\end{aligned}
$$

The first equality follows from equality (4.15). The first inequality holds by Lemma 4.3.2. The second equality follows from the definition of the downstream incremental solution $DI(v)$. The second inequality by Lemma 4.3.3, and the last inequality by Lemma 4.3.4. □

In Example 4.3.2 an RSA-game is provided and the downstream incremental solution is computed.

Example 4.3.2. Consider the RSA problem (N, e, d, r) where $N = \{1, 2, 3, 4\}, e = (2, 1.5, 1, 0), d = (3, 7.5, 1, 8.5), r_1(z) = 12z - 2z^2, r_2(z) = 12z - \frac{1}{2}z^2, r_3(z) = 13z - z^2$, and $r_4(z) = 10z$. For the coalitional values of the corresponding RSA-game we refer to the following table.

S	1	2	3	4	1,2	1,3	1,4	2,3	2,4	3,4
$v(S)$	16	16.875	12	0	37.1	28	16	28.875	16.875	12

S	1,2,3	1,2,4	1,3,4	2,3,4	N
$v(S)$	49.1	37.5	28	28.875	49.5

An optimal assignment x for N is given by $x = (0.5, 2, 1, 1)$ and displayed in Figure 4.5. The downstream incremental solution is given by

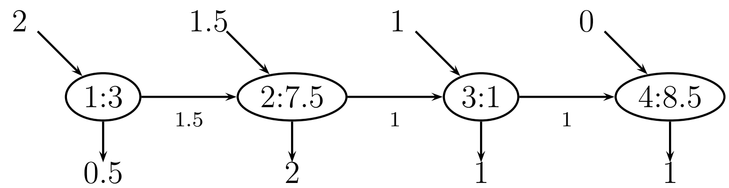

Figure 4.5: Optimal assignments x for Example 4.3.2

$$DI(v) = (16, 21.1, 12, 0.4) \in C(v)$$

Here, for example, $DI_4(v) = v(N) - v(\{1, 2, 3\}) = 0.4$.

Although the downstream incremental solution $DI(v)$ is in the core of the RSA-game v, it is a rather unsatisfying allocation, especially for upstream players. For example, 16 is allocated to player 1 according to $DI(v)$, but also $r_1(e_1) = 16$. Hence, he is not compensated for transferring 1.5 of his inflow of 2 to downstream players. $\diamondsuit$

Example 4.3.2 already indicates a possible drawback of the downstream incremental solution. Moreover, the allocations in the core are stable, but stability may contradict the TIBS principle[7]. For this, note that an allocation is stable if no coalition has an incentive to split off the grand coalition; the TIBS principle states that the water of a shared watercourse belongs to all players combined, such that coalitional deviations are not relevant in measuring the quality of an allocation. Therefore, we propose a new allocation based on the methodology of Section 4.2.

Example 4.3.3. Reconsider Example 4.3.2. From Methodology 4.2.3, the precise contribution of each player can be understood in a finite number of well specified steps. Table 4.3 provides the initial assignment and, per iteration, the assignment, the agents involved in the transfer for this assignment, and the extra total reward.

Now consider allocation $\psi(N, e, d, r)$ such that for each step of Methodology 4.2.3, the extra joint reward is evenly allocated among the participating

[7]The TIBS principle is outlined in Section 4.1

It.	Assignment	Transfer	Marg. Total Reward	Methodology based solution 1	2	3	4
0	$(2, 1\frac{1}{2}, 1, 0)$		$44\frac{7}{8}$	16	$16\frac{7}{8}$	12	0
1	$(\frac{7}{10}, 2\frac{8}{10}, 1, 0)$	$1 \to 2$	$4\frac{9}{40}$	$2\frac{9}{80}$	$2\frac{9}{80}$		
2	$(\frac{1}{2}, 2, 1, 1)$	$1, 2 \to 4$	$\frac{2}{5}$	$\frac{2}{15}$	$\frac{2}{15}$		$\frac{2}{15}$
			$49\frac{1}{2}$	$18\frac{59}{240}$	$19\frac{29}{240}$	12	$\frac{2}{15}$

Table 4.3: Methodology based solution of Example 4.3.2

players. For example, in the last step, an extra total joint reward is generated of $\frac{2}{5}$, which is evenly allocated among players $\{1, 2, 4\}$, i.e. in this iteration agents 1 and 2 transfer water to agent 4. Hence, these players are allocated $\frac{2}{15}$. For the initial assignment, each agent $i \in N$ is allocated $r_i(e_i)$. Consequently,

$$\psi(N, e, d, r) = \left(18\frac{59}{240}, 19\frac{29}{240}, 12, \frac{2}{15}\right).$$

Note that $\Psi(N, e, d, r) \in C(v)$. $\Diamond$

The following example illustrates that the methodology based solution as described above is not necessarily in the core of the corresponding game.

Example 4.3.4. Reconsider Example 4.2.2 with optimal assignment $x = (2, 0, 0, 0, 1, 4, 1, 8.5)$. Table 4.4 summarizes, per iteration, the extra total joint reward and provides the methodology based solution. Consequently,

$$\psi(N, e, d, r) = \left(36\frac{39}{40}, 11\frac{39}{40}, 14\frac{1057}{1440}, 17\frac{337}{1440}, 33\frac{535}{576}, 31\frac{97}{120}, 12, 8\frac{331}{960}\right).$$

Since $\sum_{j=3}^{8} \psi_j(N, e, d, r) = 118\frac{1}{20}$ and $v(\{3, \dots, 8\}) = 119\frac{1}{2}$, is not a core-element. As a comparison, the downstream incremental solution equals

$$DI(N, e, d, r) = \left(32, 7\frac{1}{2}, 5\frac{5}{6}, 5\frac{19}{30}, 17\frac{49}{120}, 52\frac{1}{2}, 12, 34\frac{1}{8}\right).$$

Note that Table 4.4 can also be used to compute $DI(N, e, d, r)$ by allocating, per iteration, the marginal total reward to the most downstream, i.e. receiving, agent. $\Diamond$

It.	Marg. Total Reward	Methodology Based Solution							
		1	2	3	4	5	6	7	8
0	$95\frac{1}{24}$	32	7	$5\frac{5}{6}$	$5\frac{1}{3}$	16	$16\frac{7}{8}$	12	0
1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$						
2	$\frac{3}{10}$			$\frac{3}{20}$	$\frac{3}{20}$				
3	$1\frac{49}{120}$			$\frac{169}{360}$	$\frac{169}{360}$	$\frac{169}{360}$			
4	$33\frac{1}{8}$			$8\frac{9}{32}$	$8\frac{9}{32}$	$8\frac{9}{32}$	$8\frac{9}{32}$		
5	$2\frac{1}{2}$				$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$		
6	$6\frac{1}{2}$				$2\frac{1}{6}$	$2\frac{1}{6}$			$2\frac{1}{6}$
7	$\frac{23}{32}$					$\frac{23}{64}$			$\frac{23}{64}$
8	$3\frac{9}{32}$					$1\frac{3}{32}$	$1\frac{3}{32}$		$1\frac{3}{32}$
9	$23\frac{5}{8}$	$4\frac{29}{40}$	$4\frac{29}{40}$			$4\frac{29}{40}$	$4\frac{29}{40}$		$4\frac{29}{40}$
	167	$36\frac{39}{40}$	$11\frac{39}{40}$	$14\frac{1057}{1440}$	$17\frac{337}{1440}$	$33\frac{535}{576}$	$31\frac{97}{120}$	12	$8\frac{331}{960}$

Table 4.4: Extra total joint reward and set of involved players per step of Example 4.2.2

Chapter 5

Family Sequencing and Cooperation

5.1 Introduction

Scheduling is about the optimal planning of processing a number of jobs through a number of machines. These economies of scale are fundamental to manufacturing operations. With respect to scheduling, this phenomenon manifests itself in efficiencies gained from grouping jobs together. In particular, so-called family scheduling problems have received considerable attention in the scheduling literature with setup considerations. These problems consider situations where the jobs can be classified into distinct families with respect to their production requirements such as the required tooling or container size. For the optimal planning various aspects of scheduling are taking into account. A first consideration is the objective of the optimization such as earliness (Wan and Yen (2009)), flow time (Mosheiov and Oron (2008)), or total completion time. Ahn and Hyun (1990), Bruno and Sethi (1978), Mason and Anderson (1991), and Monma and Potts (1989) propose algorithms for minimizing total weighted flow time on a single machine with family setup times. We refer to Potts and Kovalyov (2000) for a review of scheduling literature on family scheduling problems. Further we note here that sequence-dependent setup times tend to make solutions difficult to find. We refer to Allahverdi et al. (1999) and Allahverdi et al. (2008) for a review of the scheduling literature with sequence-dependent and sequence-independent setup considerations.

In this chapter, which is based on Grundel et al. (2013), we restrict attention to setup considerations of the following type. A job does not require a setup when following another job from the same family, but a "family setup time" is required when it follows a member of another family. An example of a specific application of this type of family scheduling problems is a production line of colored plastics (cf. Potts and Van Wassenhove (1992)). In this setting customer orders can be divided into color groups. A setup is required when switching from a job of one color to a job of another color. Some single machine scheduling models consider controllable processing times (Koulamas et al. (2010),Gordon and Strusevich (2009), Ocetkiewicz (2010)). To some extent, these models resemble models with family setups. Hence, if the manufacturer decides to schedule a job after a family member, its processing time does not include a setup.

Furthermore our framework assumes that per family, a cost function is defined that depends linearly on the completion time of its members. Common resources usually operate under a centralized sequencing rule (e.g., first in first out (FIFO)). Therefore, contrary to what is common in scheduling literature, we assume that there is an initial processing order σ_0 on the jobs which provides the initial right of each job to be completed at a certain time with a given set of preceding jobs. From this initial order and the cost functions, the total joint completion costs can be computed. The objective in our model is to minimize these costs.

Santos and Magazine (1985) analyzed single machine batching situations with item availability. They show that for each family, an urgency index can be computed such that, if the jobs are processed in an order of non-increasing urgency indices, then the total costs are minimized. This result is applicable to family sequencing situations, since the same objective of minimizing the total joint completion costs is considered. This result however, cannot be applied when considering minimizing the total completion costs of subgroups of jobs since the initial order on jobs puts additional constraints on the order of jobs within a subgroup. An order is admissible if each job outside the subgroup is completed at least as early as in the initial order, and its set of preceding jobs remains unchanged. In this chapter we show that for each subgroup there is an optimal order (minimizing the total joint completion costs of jobs in the subgroup) which (within components) processes all jobs of the same family consecutively, but which is not necessarily the urgency order of Santos and Magazine (1985).

After the optimal assignment is obtained, the question arises how the cost savings should be allocated among the jobs. To analyze this allocation problem, we define a cooperative family sequencing game, corresponding to the family sequencing situation, which explicitly takes into account the maximal cost savings for any subgroup of jobs. The game theoretic analysis of cost allocation problems arising from sequencing situations dates back to Curiel et al. (1989) in the setting of one machine sequencing situations with a finite number of jobs, linear cost functions and an initial order. It was shown that these games are convex and hence allow for core elements, i.e., efficient allocations that cannot be improved upon by a subgroup of jobs. The following studies in this strand of literature have extended the basic model by considering ready times (Hamers et al. (1995)), due dates (Borm et al. (2002)), precedence relations (Hamers et al. (1995)) and controllable processing times (Van Velzen (2006)). Klijn and Sánchez (2006) considered sequencing games without any initial ordering of jobs. The current chapter is one of the first to explicitly incorporate setup times. Lohmann et al. (2014) analyzes sequencing situations where some setup is required for each job which depends on its predecessor.

We show that for our class of family sequencing games, the marginal vector which corresponds to the initial order belongs to the core of the game. We also show that these games in general are not convex, σ_0-component additive, or permutationally convex (Granot and Huberman (1982)) with respect to the initial order. Therefore, the proof of the result above does not rely on standard techniques, but requires a tailor made analysis.

Finally, we specifically analyze the case where the initial order of jobs is such that all members of the same family are processed consecutively. In this case it turns out that all subgames in which the last job with respect to the initial order is not participating, are convex. From this we are able to derive a core element for the corresponding family sequencing game, based on the Shapley value (Shapley (1953)).

The outline of the chapter is as follows. Section 5.2 formally describes family sequencing situations and analyzes the optimization problem of all subgroups. With respect to the associated cost allocation problem, Section 5.3 shows that family sequencing games have a nonempty core. In Section 5.4, the specific case of ordered family sequencing is analyzed.

5.2 Family Sequencing Situations

In this section, we consider a one machine sequencing situation in which a finite number of jobs are queued in front of a machine, waiting to be processed. The machine can handle at most one job at a time. The set of jobs is denoted by N. The jobs can be partitioned into families with respect to their production requirements. Let F be the set of families. A family function $\mathcal{F} : N \to F$ associates to each job $i \in N$ the family $\mathcal{F}(i)$ that it belongs to. We denote by n_k the number of jobs in family k.

An *order on the set of jobs* is a bijection $\sigma : N \to \{1, ..., |N|\}$. We denote the set of all orders on N by $\Pi(N)$. Given an order $\sigma \in \Pi(N)$ the set of predecessors of a job $i \in N$ with respect to σ is defined as $P(\sigma, i) = \{j \in N | \sigma(j) < \sigma(i)\}$. Similarly, the set of successors of i with respect to σ is defined as $S(\sigma, i) = \{j \in N | \sigma(j) > \sigma(i)\}$. Moreover, let $\bar{P}(\sigma, i) = P(\sigma, i) \cup \{i\}$.

It is assumed that there is an initial order σ_0 on the jobs before the processing of the machine starts. If a job in family k follows a job of the same family, then it does not require a setup. However, the family setup time $s_k > 0$ is required if it is preceded by a job of a different family or if it is the first job. Observe that the setup times are independent of the family of the preceding job. We assume that each job of the same family requires the same processing time which is denoted by $p_k > 0$ for every family $k \in F$. For each job $i \in N$, the costs $c_i(t)$ of spending time t in the system is assumed to be linear in the completion time. We assume that all jobs of family k have the same cost parameter $\alpha_k > 0$ such that $c_i(t) = \alpha_k t$ for all $i \in \mathcal{F}^{-1}(k)$.

A one machine sequencing situation as described above is called a *family sequencing situation* and is denoted by $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ with $s, p, \alpha \in \mathbb{R}^F_{++}$. In a family sequencing situation the completion time $C(\sigma, i)$

of job i when processed according to the order σ is given by

$$C(\sigma, i) = \sum_{j \in \bar{P}(\sigma,i)} \left(x_{\sigma,j} s_{\mathcal{F}(j)} + p_{\mathcal{F}(j)} \right),$$

where $x_{\sigma,j}$ equals 1 if job j requires a setup when the jobs are processed with respect to σ and 0 otherwise, i.e.

$$x_{\sigma,j} = \begin{cases} 1 & \text{if } \sigma^{-1}(\sigma(j)-1) \notin \mathcal{F}(j) \text{ or } \sigma(j) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The total costs if the jobs are processed according to the order σ equals

$$\sum_{j \in N} \alpha_{\mathcal{F}(j)} C(\sigma, j).$$

By reordering the jobs with respect to σ_0, the total costs can be reduced. We call an order *optimal* if it minimizes the total costs. It was proven by Santos and Magazine (1985) and, independently, by Dobson et al. (1987) that a *highest urgency comes first* (HUCF) order is optimal for family sequencing situations. An HUCF order processes the jobs of the same family together as a group (consecutively) and processes these family groups in nonincreasing order of the family-specific urgency index u_k defined by $u_k = \frac{n_k \alpha_k}{s_k + n_k p_k}$. Here, the numerator indicates that a family with a high cost parameter is likely to be processed at the beginning of the optimal order, from the denominator can be seen that families with a high total processing time are processed in the tail of the optimal order.

Theorem 5.2.1. ***(Santos and Magazine (1985))*** *For every family sequencing situation an HUCF order is optimal.*

Example 5.2.1. Consider the family sequencing situation $(N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ with $N = \{1, 2, 3, 4, 5\}$ and $F = \{1, 2\}$. Assume that $\mathcal{F}(1) = \mathcal{F}(4) = \mathcal{F}(5) = 1$ and $\mathcal{F}(2) = \mathcal{F}(3) = 2$. Further, assume that $\sigma_0 = (1, 2, 3, 4, 5)$, $s = (3, 1)$, $p = (1, 4)$ and $\alpha = (4, 6)$. Then, the urgencies for the families are $u_1 = 2$ and $u_2 = 1\frac{1}{3}$, respectively. Hence, an HUCF order processes the jobs in family 1 first and then the jobs in family 2. In Figure 5.1, we depict the processing orders σ_0 and an HUCF order σ_N.

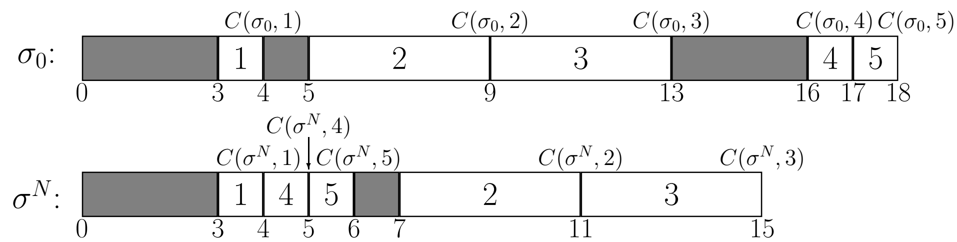

Figure 5.1: The orders σ_0 and σ_N.

The cost savings obtained by N when using σ_N equals

$$\begin{aligned} & \sum_{j\in N} \alpha_{\mathcal{F}(j)}(C(\sigma_0, j) - C(\sigma_N, j)) \\ = \ & 4(4-4) + 6(9-11) + 6(13-15) + 4(17-5) + 4(18-6) = 72. \end{aligned}$$

$\Diamond$

For a family sequencing situation $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$, the costs of a subgroup T with respect to an order σ equals $\sum_{j\in T} \alpha_{\mathcal{F}(j)} C(\sigma, j)$. We want to determine the maximal cost savings of T when its members decide to cooperate. For this aim, we have to specify which orders are admissible for T with respect to the initial order. We assume that an order $\sigma \in \Pi(N)$ is *admissible* for a subgroup T with respect to σ_0 if it satisfies the following two conditions:

(i) $P(\sigma, i) = P(\sigma_0, i)$ for all $i \in N\backslash T$, and

(ii) $C(\sigma, i) \leq C(\sigma_0, i)$ for all $i \in N\backslash T$.

Condition (i) is the standard admissibility requirement in the sequencing literature and requires that T can achieve cost savings only by changing jobs within its σ_0-components, being the maximally connected subsets of T with respect to σ_0. However, in a family sequencing situation, a subgroup may negatively affect the jobs outside the subgroup by reordering its jobs within σ_0-components. Hence, we also adopt condition (ii) which guarantees that T cannot harm the jobs outside T. The set of admissible orders of T is denoted by $\mathcal{A}(T)$. Then, the corresponding optimization problem for T in the family sequencing situation $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ is given by

$$\min_{\sigma \in \mathcal{A}(T)} \sum_{j \in T} \alpha_{\mathcal{F}(j)} C(\sigma, j).$$

An admissible order for which the minimum is attained is called *optimal* for T. The remainder of this section is dedicated to these optimal orders.

Example 5.2.2. Reconsider the family sequencing situation from Example 5.2.1. Now consider the subgroup $S = \{1, 2, 3\}$. Observe that the order $\sigma = (2, 3, 1, 4, 5)$ is an admissible order for S with respect to σ_0: $P(\sigma, i) = P(\sigma_0, i)$ for all $i \in N \backslash S$ and both $C(\sigma, 4) \leq C(\sigma_0, 4)$ and $C(\sigma, 5) \leq C(\sigma_0, 5)$. This can be seen from Figure 5.2.

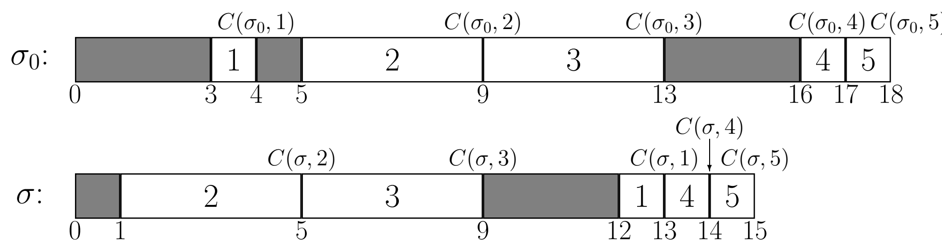

Figure 5.2: The orders σ_0 and σ.

The cost savings obtained by S when using σ equal $\sum_{j \in S} \alpha_{\mathcal{F}(j)}(C(\sigma_0, j) - C(\sigma, j)) = 12$. Actually, σ is an optimal order for S. Notice that σ processes the jobs in family 2 first although with respect to the optimal order for all jobs, jobs in family 1 are processed first.

Now, consider the subgroup $T = \{1, 2, 3, 5\}$. Clearly, σ is an admissible order for T. Actually, σ is also an optimal order for T. The cost savings obtained by T using σ equals $\sum_{j \in T} \alpha_{\mathcal{F}(j)}(C(\sigma_0, j) - C(\sigma, j)) = 12 + 4(18 - 15) = 24$. That is, when the jobs 1,2 and 3 reorder themselves from σ_0 to σ, also job 5 profits from an earlier completion time and this profit is now taken into account. $\diamondsuit$

Let $T \subset N$ and $\sigma \in \Pi(N)$ be such that T is a connected subgroup with respect to σ. Define,

$$f(\sigma, T) = \arg\min_{j \in T} \sigma(j),$$

and

$$l(\sigma, T) = \arg\max_{j \in T} \sigma(j).$$

Clearly, $f(\sigma, T)$ is the first job within T, and $l(\sigma, T)$ the last job within T with respect to the order σ. We denote by $P(\sigma, T)$ the collection of jobs which precede every member of T in the order σ, i.e.,

$$P(\sigma, T) = \{j \in N | \sigma(j) < \sigma(f(\sigma, T))\}$$

and $S(\sigma, T)$ the collection of jobs which succeed every member of T in the order σ,

$$S(\sigma, T) = \{j \in N | \sigma(j) > \sigma(l(\sigma, T))\}.$$

Let $\sigma \in \Pi(N)$. We call a set of jobs that are processed between two setups when the jobs are processed with respect to σ, a *run of* σ. Obviously, all jobs in the same run are of the same family. A run which consists of jobs of family k is called a *run of family* k.

An order σ is *family ordered* if it processes all jobs that belong to the same family consecutively, i.e. if for every pair of jobs $i, j \in N$ where $\mathcal{F}(i) = \mathcal{F}(j)$ it holds that $h \in \mathcal{F}(i)$ for every h with $\sigma(i) < \sigma(h) < \sigma(j)$.

Since a σ_0-component of a subgroup can affect the completion times of the members of another σ_0-component behind it (cf. Example 5.2.2), it is generally not easy to find an optimal admissible order for a subgroup. Nevertheless, there are useful properties regarding the structure of optimal admissible orders. In the following theorem we show that for every connected subgroup, there exist an optimal admissible order that processes the jobs of the same family consecutively.

Theorem 5.2.2. *Let* $(N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ *be a family sequencing situation and let* $T \subset N$ *be a subset of jobs. Then, there exists an optimal order for* T *which processes all jobs of the same family within a* σ_0*-component of* T *consecutively.*

Proof. Let $T = T_1 \cup T_2 \cup \cdots \cup T_l$ where for each $y \in \{1, \ldots, l\}$, T_y is a maximally connected subset of T with respect to σ_0. Let $\sigma \in \mathcal{A}(T)$ and suppose σ is an optimal order for T.

Fix $y \in \{1, 2, ..., l\}$ and suppose that with respect to σ, family k jobs in T_y are processed in different runs. Let K_1 and K_2 be the set of family k jobs in T_y that belong to the first and second run, respectively. Let M be the set of jobs (of other families) that are placed in between K_1 and K_2 with respect to σ. Let γ be the time to process and setup all jobs in M when they are processed with respect to σ, i.e., $\gamma = \sum_{j \in M}(x_{\sigma,j} s_{\mathcal{F}(j)} + p_{\mathcal{F}(j)})$. Let $i_1 = f(\sigma, K_1)$, $i_2 = f(\sigma, K_2)$, $m = f(\sigma, M)$, and $h = f(\sigma, S(\sigma, K_2))$.

Now consider the order $\sigma' \in \Pi(N)$ which is obtained from σ by moving all jobs in K_1 to the head of K_2 and the order $\sigma'' \in \Pi(N)$ which is obtained from σ by moving all jobs in K_2 to the tail of K_1. Figure 5.3 depicts the orders σ, σ' and σ''.

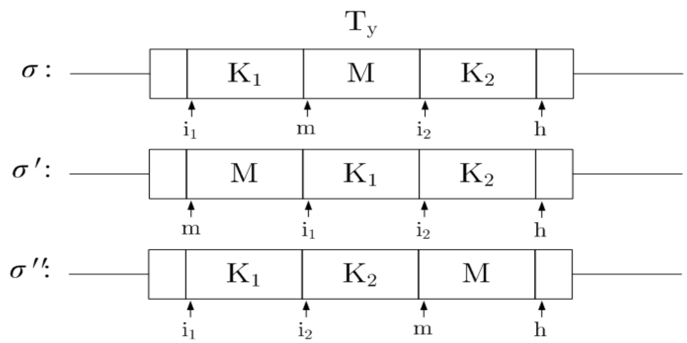

Figure 5.3: The orders σ, σ' and σ''.

We start by showing that σ' and σ'' are admissible. Next we show that either σ' or σ'' is also an optimal order, i.e.

$$\sum_{j \in T} \alpha_{\mathcal{F}(j)}(C(\sigma, j) - C(\sigma', j)) \geq 0 \tag{5.1}$$

or

$$\sum_{j \in T} \alpha_{\mathcal{F}(j)}(C(\sigma, j) - C(\sigma'', j)) \geq 0. \tag{5.2}$$

First, we show that σ' is an admissible order for T. Clearly, $P(\sigma', i) = P(\sigma, i) = P(\sigma_0, i)$ for all $i \in N \backslash T$. It remains to show that $C(\sigma', i) \leq C(\sigma_0, i)$ for every $i \in N \backslash T$. Since σ is an admissible order, it is sufficient to show that $C(\sigma', i) \leq C(\sigma, i)$ for every $i \in N \backslash T$. Observe that $x_{\sigma',j} = x_{\sigma,j}$ for every

$j \in N\backslash\{i_1, i_2, m\}$. Hence, $C(\sigma', i) = C(\sigma, i)$ for every $i \in P(\sigma, K_1)$ and

$$\begin{aligned} C(\sigma,i) - C(\sigma',i) &= \sum_{j\in\bar{P}(\sigma,i)} \big(s_{\mathcal{F}(j)} x_{\sigma,j} + p_{\mathcal{F}(j)}\big) - \sum_{j\in\bar{P}(\sigma',i)} \big(s_{\mathcal{F}(j)} x_{\sigma',j} + p_{\mathcal{F}(j)}\big) \\ &= \sum_{j\in\{i_1,i_2,m\}} (x_{\sigma,j} - x_{\sigma',j})\, s_{\mathcal{F}(j)} \\ &\geq 0, \end{aligned}$$

for every $i \in S(\sigma, M)$. The inequality follows from $x_{\sigma,m} = 1$ which implies $(x_{\sigma,m} - x_{\sigma',m}) s_{\mathcal{F}(m)} \geq 0$. Further it holds that $x_{\sigma,i_2} = 1$, $x_{\sigma',i_1} + x_{\sigma',i_2} = 1$, and $s_{\mathcal{F}(i_1)} = s_{\mathcal{F}(i_2)}$ such that $\sum_{j\in\{i_1,i_2\}} (x_{\sigma,j} - x_{\sigma',j})\, s_{\mathcal{F}(j)} = x_{\sigma,i_1} s_{\mathcal{F}(i_1)} \geq 0$. Hence, $\sigma' \in \mathcal{A}(T)$.

Next we show that σ'' is an admissible order for T. Obviously, $P(\sigma'', i) = P(\sigma_0, i)$ for every $i \in N\backslash T$. Since σ is an admissible order, it is sufficient to show that $C(\sigma'', i) \leq C(\sigma, i)$ for every $i \in N\backslash T$. It can be observed that $x_{\sigma'',i} = x_{\sigma,i}$ for every $i \in N\backslash\{i_2, h\}$. Hence, $C(\sigma'', i) = C(\sigma, i)$ for every $i \in P(\sigma, M)$ and

$$\begin{aligned} C(\sigma,i) - C(\sigma'',i) &= \sum_{j\in\bar{P}(\sigma,i)} \big(s_{\mathcal{F}(j)} x_{\sigma,j} + p_{\mathcal{F}(j)}\big) \\ &\quad - \sum_{j\in\bar{P}(\sigma'',i)} \big(s_{\mathcal{F}(j)} x_{\sigma'',j} + p_{\mathcal{F}(j)}\big) \\ &= \sum_{j\in\{i_2,h\}} (x_{\sigma,j} - x_{\sigma'',j}) s_{\mathcal{F}(j)}, \end{aligned}$$

for every $i \in S(\sigma, K_2)$. Clearly, $x_{\sigma,i_2} = 1$ and $x_{\sigma'',i_2} = 0$. However, $x_{\sigma,h}$ can either be 0 or 1. First assume that $x_{\sigma,h} = 0$, i.e., h is member of family k. Hence, $x_{\sigma'',h} = 1$ and

$$\sum_{j\in\{i_2,h\}} (x_{\sigma,j} - x_{\sigma'',j}) s_{\mathcal{F}(j)} = (1-0) s_{\mathcal{F}(i_2)} + (0-1) s_{\mathcal{F}(h)} = (1-0) s_k + (0-1) s_k = 0.$$

Next assume that $x_{\sigma,h} = 1$. Then,

$$\begin{aligned} \sum_{j\in\{i_2,h\}} (x_{\sigma,j} - x_{\sigma'',j}) s_{\mathcal{F}(j)} &= (1-0) s_k + (1 - x_{\sigma'',h}) s_{\mathcal{F}(h)} \\ &\geq (1-0) s_k + (1-1) s_{\mathcal{F}(h)} = s_k. \end{aligned}$$

Hence $C(\sigma, i) - C(\sigma'', i) \geq 0$ for every $i \in S(\sigma, K_2)$. This yields that $\sigma'' \in \mathcal{A}(T)$.

It remains to prove that either (5.1) or (5.2) is satisfied. First, observe that

$$\sum_{j\in T}\alpha_{\mathcal{F}(j)}(C(\sigma,j)-C(\sigma',j)) \geq \sum_{j\in K_1\cup M}\alpha_{\mathcal{F}(j)}\left(C(\sigma,j)-C(\sigma',j)\right), \quad (5.3)$$

where the inequality follows from the fact that $C(\sigma',i)=C(\sigma,i)$ for every $i\in P(\sigma,K_1)$ and $C(\sigma,i)-C(\sigma',i)\geq 0$ for every $i\in S(\sigma,M)$. Moreover, for every $i\in M$ it holds that

$$\begin{aligned} C(\sigma,i)-C(\sigma',i) &= \sum_{j\in K_1}(x_{\sigma,j}s_{\mathcal{F}(j)}+p_{\mathcal{F}(j)})+(x_{\sigma,m}-x_{\sigma',m})s_{\mathcal{F}(m)} \\ &= x_{\sigma,i_1}s_k+|K_1|p_k+(x_{\sigma,m}-x_{\sigma',m})s_{\mathcal{F}(m)} \\ &= x_{\sigma,i_1}s_k+|K_1|p_k+(1-x_{\sigma',m})s_{\mathcal{F}(m)} \\ &\geq |K_1|p_k, \end{aligned} \quad (5.4)$$

and for every $i\in K_1$

$$\begin{aligned} C(\sigma,i)-C(\sigma',i) &= -\sum_{j\in M}(x_{\sigma',j}s_{\mathcal{F}(j)}+p_{\mathcal{F}(j)})-(x_{\sigma',i_1}-x_{\sigma,i_1})s_k \\ &\geq -(\gamma+s_k). \end{aligned} \quad (5.5)$$

The inequality follows from $x_{\sigma',i_1}=1$ and

$$\sum_{j\in M}(x_{\sigma',j}s_{\mathcal{F}(j)}+p_{\mathcal{F}(j)}) = \begin{cases} \gamma, & \text{if } x_{\sigma',m}=1, \\ \gamma-s_{\mathcal{F}(m)}, & \text{if } x_{\sigma',m}=0. \end{cases}$$

Consequently, by inequalities (5.3)-(5.5), we have that

$$\sum_{j\in T}\alpha_{\mathcal{F}(j)}\left(C(\sigma,j)-C(\sigma',j)\right) \geq |K_1|\left(p_k\sum_{j\in M}\alpha_{\mathcal{F}(j)}-(\gamma+s_k)\alpha_k\right). \quad (5.6)$$

Secondly, observe that

$$\sum_{j\in T}\alpha_{\mathcal{F}(j)}(C(\sigma,j)-C(\sigma'',j)) \geq \sum_{j\in K_2\cup M}\alpha_{\mathcal{F}(j)}\left(C(\sigma,j)-C(\sigma'',j)\right),$$

which follows from $C(\sigma'',i)=C(\sigma,i)$ for every $i\in P(\sigma,M)$ and $C(\sigma,i)-C(\sigma'',i)\geq 0$ for every $i\in S(\sigma,K_2)$. Moreover,

$$C(\sigma,i)-C(\sigma'',i) = \begin{cases} \gamma+s_k, & \text{if } i\in K_2, \\ -|K_2|p_k, & \text{if } i\in M, \end{cases}$$

which implies that

$$\sum_{j \in T} \alpha_{\mathcal{F}(j)}(C(\sigma, j) - C(\sigma'', j)) \geq |K_2| \left((\gamma + s_k)\alpha_k - p_k \sum_{j \in M} \alpha_{\mathcal{F}(j)} \right). \quad (5.7)$$

The right-hand side in either (5.6), or (5.7) is non-negative, which shows that either (5.1) or (5.2) is satisfied. □

From Theorem 5.2.2, it follows that for the optimization problem for a subgroup of jobs, the urgency indices of the jobs are not the only factor to take into consideration. Apparently, the structure of families within the subgroup is also of concern. Therefore, the *family urgency index* for subgroups of agents is introduced.

Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be a family sequencing situation and let T be a connected subset with respect to σ_0. Let $\mathcal{F}(T) = \bigcup_{i \in T} \mathcal{F}(i)$ be the set of families associated to T. For $k \in \mathcal{F}(T)$, let $n_{T,k}$ be the number of jobs of family k that belongs to T. The family urgency index $u_{T,k}$ is defined as

$$u_{T,k} = \frac{n_{T,k}\alpha_k}{s_k + n_{T,k}p_k}.$$

Now, assume that $\mathcal{F}(l(\sigma_0, T)) = \bar{k}(T)$. The *tail-adjusted family urgency index* $u'_{T,l}$ for T is defined as

$$u'_{T,l} = \begin{cases} u_{T,l} & \text{if } l \neq \bar{k}(T), \\ 0 & \text{if } l = \bar{k}(T). \end{cases}$$

for all $l \in \mathcal{F}(T)$. Hence, the family of the last job in T (with respect to σ_0) obtains the lowest urgency index.

An order $\sigma \in \Pi(N)$ is called *an HUCF order for* T, where T is connected with respect to σ_0, if

(i) $P(\sigma, i) = P(\sigma_0, i)$ for every $i \in N \backslash T$, and

(ii) σ is family ordered and processes the family groups in non-increasing order of the family urgency index for T.

An order $\sigma \in \Pi(N)$ is called *a tail-adjusted HUCF order for* T, where T is connected with respect to σ_0, if

(i) $P(\sigma, i) = P(\sigma_0, i)$ for every $i \in N \backslash T$, and

(ii) σ is family ordered and processes the family groups in non-increasing order of the tail-adjusted family urgency index for T.

This yields that the family members of the last job in T (with respect to σ_0) are processed last.

In the following lemma, we focus on the structure of the optimal orders for connected subgroups which include the job that is processed first with respect to σ_0. We show that for these subgroups, either an HUCF order or a tail-adjusted HUCF order is optimal. IF HUCF order is not admissible for T, then this is because additional setup time is induced for the first job after T. In that case, the tail-adjusted HUCF order keep the last job of T equal to the initial order σ_0.

Lemma 5.2.3. *Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be a family sequencing situation and T be a connected subgroup with respect to σ_0 with $\sigma_0^{-1}(1) \in T$. Then,*

(i) if an HUCF order for T is admissible, then it is optimal for T.

(ii) if an HUCF order for T is not admissible, then a tail-adjusted HUCF order for T is optimal for T.

Proof. Let $\sigma \in \Pi(N)$ be an HUCF order for T.

(i) Let $\sigma \in \mathcal{A}(T)$. From Theorem 5.2.1 and Theorem 5.2.2 it immediately follows that σ is optimal.

(ii) Let $\sigma \notin \mathcal{A}(T)$. Then, clearly, $S(\sigma_0, T) \neq \emptyset$. Define $h = f(\sigma_0, S(\sigma_0, T))$. We first prove the following claim:

Claim: Let $\pi \in \Pi(N)$ be such that $P(\pi, i) = P(\sigma_0, i)$ for all $i \in N \setminus T$, π is family ordered for T, and $\pi \notin \mathcal{A}(T)$. Then $C(\pi, h) > C(\sigma_0, h)$, $x_{\pi,h} = 1$ and $x_{\sigma_0,h} = 0$.

Proof of the claim: Since $\pi \notin \mathcal{A}(T)$ and $P(\pi, i) = P(\sigma_0, i)$ for all $i \in N \setminus T$, it holds that

$$C(\pi, i) > C(\sigma_0, i), \tag{5.8}$$

for some $i \in S(\pi, T)$. Since $\sigma_0^{-1}(1) \in T$ and since π is family ordered, the number of setups in T are minimized, which implies the total setup time is minimized, and, consequently

$$C(\pi, l(\pi, T)) \leq C(\sigma_0, l(\sigma_0, T)).$$

Hence, since π and σ_0 coincide after the last job in T, for (5.8) to hold it must be the case that $x_{\pi,h} = 1$ and $x_{\sigma_0,h} = 0$, while in fact $C(\pi, h) > C(\sigma_0, h)$. This proves the claim. Clearly, the claim implies that

$$C(\sigma, h) > C(\sigma_0, h),\ x_{\sigma,h} = 1,\ \text{and } x_{\sigma_0,h} = 0. \tag{5.9}$$

Let $\sigma' \in \Pi(N)$ denote a tail-adjusted HUCF order for T. Since σ' is family ordered, $P(\sigma', i) = P(\sigma_0, i)$ for all $i \in N \setminus T$, and $x_{\sigma',h} = 0$, we readily see from the claim that $\sigma' \in \mathcal{A}(T)$.

Now consider an arbitrary order $\tau \in \Pi(N)$ with $\tau \in \mathcal{A}(T)$. We prove that all jobs in T that are within $\mathcal{F}(h)$ are processed last in T. From (5.9), it follows that $T \cap \mathcal{F}(h) \neq \emptyset$. Now suppose $x_{\tau,h} = 1$ or that at least two members of $\mathcal{F}(h)$ within T are processed in different runs. Then,

$$\begin{aligned} C(\tau, h) &\geq \sum_{k \in \mathcal{F}(T)} (s_k + n_{T,k} p_k) + s_{\mathcal{F}(h)} + p_{\mathcal{F}(h)} \\ &= C(\sigma, h) \\ &> C(\sigma_0, h). \end{aligned}$$

The equality holds by the fact that σ is an HUCF order and $x_{\sigma,h} = 1$. The strict inequality follows from (5.9). This establishes a contradiction with the admissibility of τ. Hence, for each admissible order there is no setup required for job h, and the members of $\mathcal{F}(h)$ are not processed in different runs in T. This implies that for each admissible order all members of $\mathcal{F}(h)$ are sent to the back of T. The tail-adjusted HUCF order σ' satisfies this condition.

It remains to prove that σ' is optimal for the set $T' = \{i \in T | \mathcal{F}(i) \neq \mathcal{F}(h)\}$. This is obvious by applying *(i)* since σ' is defined as an HUCF order for T' which is admissible for T'. $\square$

The following example illustrates Lemma 5.2.3.

Example 5.2.3. Reconsider the family sequencing situation from Example 5.2.2. Consider $S = \{1, 2\}$. Then, $\mathcal{F}(S) = \{1, 2\}$ and $u_{S,1} = 1$ and $u_{S,2} = 1\frac{1}{5}$, respectively. Since $\bar{k}(S) = 2$, the tail-adjusted urgencies in S are $u'_{S,1} = 1$ and $u'_{S,2} = 0$, respectively. Hence, $\sigma = (2, 1, 3, 4, 5)$ is an HUCF order and $\sigma' = \sigma_0 = (1, 2, 3, 4, 5)$ is a tail-adjusted HUCF order for S.

By the fact that σ is not admissible for S, i.e. $\sigma \notin \mathcal{A}(S)$, the tail-adjusted order σ' is optimal for S. By $\sigma' = \sigma_0$ it holds that $\sum_{j \in T} \alpha_{\mathcal{F}(j)}(C(\sigma_0, j) - C(\sigma', j)) = 0$. $\diamondsuit$

5.3 Family Sequencing Games

A *transferable utility* (TU) *game* is an ordered pair (N, v) where N is the finite set of players, and v the characteristic function on 2^N, the collection of all subsets of N. The function v assigns to every *coalition* $T \in 2^N$ a real number $v(T)$, with $v(\emptyset) = 0$. As far as it concerns *family sequencing games*, we denote a *subgroup* of jobs T, by *coalition* T. Here, $v(T)$ is called the worth or value of coalition T. The set of all TU-games with player set N is denoted by TU^N. Where no confusion arises, we write v rather than (N, v). A game v is called *monotonic* if $v(S) \leq v(T)$ for every $S \subset T$ and v is called *superadditive* if $v(S) + v(T) \leq v(S \cup T)$ for every $S, T \in 2^N$ with $T \cap S = \emptyset$. A game v is *convex* if a player's marginal contribution does not decrease if he joins a larger coalition, i.e., $v(T \cup \{i\}) - v(T) \geq v(S \cup \{i\}) - v(S)$ for every $i \in N$ and $S, T \subset N \backslash \{i\}$ with $S \subset T$.

The *core* of a game v, denoted by $Core(v)$, is defined as the set of efficient allocations for which no coalition has an incentive to split off from the grand coalition, i.e.,

$$Core(v) = \left\{ x \in \mathbb{R}^N \middle| \sum_{j \in N} x_j = v(N) \text{ and } \sum_{j \in S} x_j \geq v(S) \text{ for all } S \in 2^N \right\}.$$

A game with a nonempty core is called *balanced*.

A coalition $S \subset N$ is called *connected with respect to an order* $\sigma \in \Pi(N)$ if for all $i, j \in S$ and $h \in N$ such that $\sigma(i) < \sigma(h) < \sigma(j)$ it holds that $h \in S$. For a coalition S, $S \backslash \sigma$ denotes the set of σ-components of S.

Let $\sigma \in \Pi(N)$. A TU-game v is called *σ-component additive* if it satisfies the following three conditions:

(i) $v(\{i\}) = 0$ for all $i \in N$,

(ii) v is superadditive, and

(iii) $v(S) = \sum_{T \in S\backslash\sigma} v(T)$ for all $S \in 2^N$.

Le Breton et al. (1992) showed that σ-component additive games are balanced.

In a *family sequencing game* corresponding to a family sequencing situation, players will correspond to jobs, and the value of a coalition T is defined as the maximum cost savings that coalition T can achieve by means of an admissible order in $\mathcal{A}(T)$. Formally, the family sequencing game v corresponding to a family sequencing situation $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ is defined by

$$v(T) = \max_{\sigma \in \mathcal{A}(T)} \left\{ \sum_{j \in T} \alpha_{\mathcal{F}(j)} \ (C(\sigma_0, j) - C(\sigma, j)) \right\}$$

for every $T \subset N$. It readily follows that v is monotonic.

Example 5.3.1. Reconsider the family sequencing situation of Examples 5.2.1 - 5.2.3. Since $\sigma = (2, 3, 1, 4, 5)$ is an optimal order for $S = \{1, 2, 3\}$ and $T = \{1, 2, 3, 5\}$, it holds, for the corresponding family sequencing game v, that $v(S) = 12$ and $v(T) = 24$. From this, one can see that the game is not σ_0-component additive; $\{1, 2, 3\}$ and $\{5\}$ are the σ_0-components of $\{1, 2, 3, 5\}$, but $v(\{1, 2, 3, 5\}) = 24 \neq 12 = v(\{1, 2, 3\}) + v(\{5\})$.

The complete family sequencing game v is given by: $v(N) = v(\{2, 3, 4, 5\}) = 72$, $v(\{1, 2, 3, 4\}) = 59$, $v(\{1, 2, 3, 5\}) = 24$, $v(\{2, 3, 4\}) = 36$, $v(\{1, 2, 3\}) = 12$ and $v(S) = 0$ for every remaining coalition $S \in 2^N$. Also observe that this game is not convex since for $i = 1, S = \{2, 3\}$, and $T = \{2, 3, 4, 5\}$,

$$v(T \cup \{i\}) - v(T) = 0 < 12 = v(S \cup \{i\}) - v(S).$$

$\diamondsuit$

Let $v \in TU^N$. The *marginal vector* $m^\sigma(v) \in \mathbb{R}^N$ with respect to $\sigma \in \Pi(N)$ is defined, for each $i \in N$, by

$$m_i^\sigma(v) = v(\bar{P}(\sigma, i)) - v(P(\sigma, i)).$$

A concept that is closely related to convexity is permutationally convexity. The game v is *permutationally convex* with respect to $\sigma \in \Pi(N)$ if

$$v(\bar{P}(\sigma, i) \cup T) - v(\bar{P}(\sigma, i)) \leq v(\bar{P}(\sigma, j) \cup T) - v(\bar{P}(\sigma, j)),$$

for every $i, j \in N$ with $\sigma(i) < \sigma(j)$ and $T \subset S(\sigma, j)$. Permutational convexity with respect to an order $\sigma \in \Pi(N)$ is a well-known sufficient condition for the corresponding marginal vector $m^\sigma(v)$ to be a core element (cf. Granot and Huberman (1982)). In the following example we show that family sequencing games need not be permutationally convex with respect to initial order σ_0.

Example 5.3.2. Consider the family sequencing situation $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ with $N = \{1, 2, 3, 4, 5, 6\}$ and $F = \{1, 2, 3, 4\}$. Assume that $\mathcal{F}(1) = \mathcal{F}(3) = 1$, $\mathcal{F}(2) = 2$, $\mathcal{F}(4) = \mathcal{F}(5) = 3$ and $\mathcal{F}(6) = 4$. Furthermore, let $\sigma_0 = (1, 2, 3, 4, 5, 6)$, $s = (2, 2, 1, 5)$, $p = (1, 2, 2, 5)$ and $\alpha = (10, 10, 10, 1)$. Finally, let v be the family sequencing game corresponding to $\Sigma(N)$.

Consider the coalitions

$$\begin{aligned} S &= \bar{P}(\sigma_0, 3) = \{1, 2, 3\}, & S' &= \bar{P}(\sigma_0, 3) \cup \{6\} = \{1, 2, 3, 6\}, \\ W &= \bar{P}(\sigma_0, 4) = \{1, 2, 3, 4\}, & W' &= \bar{P}(\sigma_0, 4) \cup \{6\} = \{1, 2, 3, 4, 6\}. \end{aligned}$$

The urgency indices for S are $u_{S,1} = \frac{20}{4}$ and $u_{S,2} = \frac{10}{4}$. Hence, $\sigma_S = (1, 3, 2, 4, 5, 6)$ is an HUCF order for S. Clearly, σ_S is admissible for S. Then, by Lemma 5.2.3, σ_S is optimal for S.

The urgency indices for W are $u_{W,1} = \frac{20}{4}$, $u_{W,2} = \frac{10}{4}$ and $u_{W,3} = \frac{10}{3}$. Hence, $\sigma_W = (1, 3, 4, 2, 5, 6)$ is an HUCF order for W. It can easily be observed that σ_W is admissible for W. Then, by Lemma 5.2.3, σ_W is optimal for W.

Clearly, σ_S is also an optimal order for S' while σ_W is also an optimal order for W'. Then, v is not permutationally convex with respect to σ_0 since

$$2 = v(S') - v(S) > v(W') - v(W) = 1.$$

$\Diamond$

In the next theorem we will prove that for each family sequencing game, the marginal vector with respect to the initial order is stable and efficient, i.e. v is balanced. In the literature on sequencing games, balancedness of a game v is often proved by using the fact that $v \in TU^N$ is σ_0-component additive. However, in Example 5.3.1, it is shown that family sequencing games need not be σ_0-component additive. For sequencing games with controllable processing times, Van Velzen (2006) proved balancedness by using the property of permutationally convexity. From Example 5.3.2 it can be seen that family sequencing games are not permutationally convex with respect to σ_0. Hence, the balancedness of family sequencing games cannot be proved using standard techniques. However, a direct, but technically intricate proof, shows that the marginal vector corresponding to the initial order, σ_0, does belong to the core of a family sequencing game. For this proof we introduce some orders which we prove to be admissible and hence have a completion time smaller than or equal to the completion time of the initial order.

Theorem 5.3.1. *Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be a family sequencing situation and let $v \in TU^N$ be the corresponding sequencing game. Then, $m^{\sigma_0}(v) \in Core(v)$.*

Proof. Let $T \backslash \sigma_0 = \{T_1, T_2, ..., T_l\}$ be such that $T_y \subset P(\sigma_0, T_{y+1})$ for every $y \in \{1, ..., l-1\}$. Here $P(\sigma_0, Ty+1) = P(\sigma_0, f(\sigma_0, T_{y+1}))$, i.e. the set of predecessors of the first job within T_{y+1} with respect to σ_0. Let $\sigma \in \mathcal{A}(T)$ be an optimal order for T.

We have to show that $\sum_{j \in T} m_j^{\sigma_0}(v) \geq v(T)$. Since

$$\begin{aligned} v(T) &= \sum_{j \in T} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j) \right) \\ &= \sum_{y=1}^{l} \sum_{j \in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j) \right), \end{aligned}$$

and

$$\sum_{j \in T} m_j^{\sigma_0}(v) = \sum_{y=1}^{l} \sum_{j \in T_y} m_j^{\sigma_0}(v),$$

it is sufficient to show that

$$\sum_{j\in T_y} m_j^{\sigma_0}(v) \geq \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right), \tag{5.10}$$

for every $y \in \{1, 2, ..., l\}$.

Pick $y \in \{1, 2, ..., l\}$. Let $D = P(\sigma_0, T_y)$ and $E = D \cup T_y$. If $D = \emptyset$, then $y = 1$ and (5.10) follows from Lemma 5.2.3 and Theorem 5.2.2, which implies that σ is also an optimal order for T_1. Hence,

$$\sum_{j\in T_1} m_j^{\sigma_0}(v) = v(T_1) = \sum_{j\in T_1} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right).$$

For the remainder, let $D \neq \emptyset$. Notice that $\sigma_0^{-1}(1) \in D$ and $\sum_{j\in T_y} m_j^{\sigma_0}(v) = v(E) - v(D)$. Hence inequality (5.10) boils down to

$$v(E) - v(D) - \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right) \geq 0. \tag{5.11}$$

In the remainder of this proof we consider an order μ for E which is the combination of an optimal order for T_y and an optimal order for D. If μ is admissible for E, then inequality (5.11) can be shown directly. If μ is not admissible, then we construct an adjusted admissible order μ' to indirectly verify (5.11).

First, we denote by σ_y and by $\sigma_{\vec{y}}$ the orders defined by

$$\sigma_y(i) = \begin{cases} \sigma(i), & \text{if } i \in T_y, \\ \sigma_0(i), & \text{otherwise,} \end{cases}$$

$$\sigma_{\vec{y}}(i) = \begin{cases} \sigma(i), & \text{if } i \in \bigcup_{q=1}^{y} T_q, \\ \sigma_0(i), & \text{otherwise.} \end{cases}$$

Notice that $\sigma_{\vec{y}}$ is an admissible order for E. Moreover, let π be an optimal order for D. Then,

$$v(D) = \sum_{j\in D} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\pi, j)\right).$$

Since $\sigma_0^{-1}(1) \in D$, we can choose π to be either an HUCF order for D or a tail-adjusted HUCF order for D (cf. Lemma 5.2.3). Let $\mu \in \Pi(N)$ be the order defined by

$$\mu(i) = \begin{cases} \pi(i), & \text{if } i \in D, \\ \sigma_y(i), & \text{otherwise.} \end{cases}$$

Observe that $P(\sigma_0, i) = P(\mu, i)$ for every $i \in N \backslash E$.

Let $i_y = f(\sigma, T_y)$ and $k = \mathcal{F}(i_y)$. For all $i \in S(\sigma_0, D)$, observe that

$$\begin{aligned} C(\sigma_{\vec{y}}, i) - C(\mu, i) &= \sum_{j \in \bar{P}(\sigma_{\vec{y}}, i)} \left(s_{\mathcal{F}(j)} x_{\sigma_{\vec{y}}, j} + p_{\mathcal{F}(j)} \right) \\ &\quad - \sum_{j \in \bar{P}(\mu, i)} \left(s_{\mathcal{F}(j)} x_{\mu, j} + p_{\mathcal{F}(j)} \right) \\ &= \sum_{j \in D} s_{\mathcal{F}(j)} (x_{\sigma_{\vec{y}}, j} - x_{\mu, j}) + s_k (x_{\sigma_{\vec{y}}, i_y} - x_{\mu, i_y}) \\ &\geq s_k (x_{\sigma_{\vec{y}}, i_y} - x_{\mu, i_y}). \end{aligned} \tag{5.12}$$

Here the second equality follows from the fact that $\bar{P}(\sigma_{\vec{y}}, i) = \bar{P}(\mu, i)$ for every $i \in S(\sigma_0, D)$ and $x_{\sigma_{\vec{y}}, i} = x_{\mu, i}$ for every $i \in S(\sigma_0, D) \backslash \{i_y\}$. The inequality follows from the fact that, with respect to μ, the members of D are processed according to π which is an HUCF or a tail-adjusted HUCF order for D and these orders require the minimum total setup time to process the jobs in D. Hence, for all $h \in \mathcal{F}(D)$ it holds that

$$\sum_{j \in D : j \in \mathcal{F}^{-1}(h)} x_{\sigma_{\vec{y}}, j} \geq 1,$$

and

$$\sum_{j \in D : j \in \mathcal{F}^{-1}(h)} x_{\mu, j} = 1.$$

We now distinguish between two cases.

First assume $x_{\sigma_{\vec{y}}, i_y} - x_{\mu, i_y} \geq 0$. Then, μ is admissible for E. For this, first observe that $P(\sigma_0, i) = P(\mu, i)$ for every $i \in N \backslash E = S(\sigma_0, E)$. Next, by admissibility of $\sigma_{\vec{y}}$, it holds that $C(\sigma_0, i) \geq C(\sigma_{\vec{y}}, i)$ for every $i \in S(\sigma_0, E)$.

Hence, by inequality (5.12) it holds that $C(\sigma_{\vec{y}}, i) \geq C(\mu, i)$ for every $i \in S(\sigma_0, E)$. We may conclude that μ is an admissible order for E and that

$$v(E) \geq \sum_{j \in E} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\mu, j) \right). \tag{5.13}$$

Combining (5.12) and (5.13) one obtains

$$\begin{aligned}
& v(E) - v(D) - \sum_{j \in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j) \right) \\
\geq \; & \sum_{j \in E} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\mu, j) \right) - \sum_{j \in D} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\pi, j) \right) \\
& - \sum_{j \in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, i) - C(\sigma, j) \right) \\
= \; & \sum_{j \in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma, i) - C(\mu, j) \right) \\
= \; & \sum_{j \in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_{\vec{y}}, i) - C(\mu, j) \right) \\
\geq \; & s_k (x_{\sigma_{\vec{y}}, i_y} - x_{\mu, i_y}) \sum_{j \in T_y} \alpha_{\mathcal{F}(j)} \\
\geq \; & 0.
\end{aligned}$$

The first equality follows from the fact that $C(\pi, i) = C(\mu, i)$ for all $i \in D$. The second equality holds by $C(\sigma, i) = C(\sigma_{\vec{y}}, i)$ for all $i \in T_y$. The second inequality follows from (5.12). Hence, inequality (5.11) is satisfied if $x_{\sigma_{\vec{y}}, i_y} - x_{\mu, i_y} \geq 0$.

Secondly, assume $x_{\sigma_{\vec{y}}, i_y} - x_{\mu, i_y} < 0$, i.e., for the rest of the proof assume that $x_{\sigma_{\vec{y}}, i_y} = 0$ and $x_{\mu, i_y} = 1$. Since $x_{\sigma_{\vec{y}}, i_y} = 0$, $\mathcal{F}(i_y) = \mathcal{F}(l(\sigma_y, D)) = \mathcal{F}(l(\sigma_{\vec{y}}, D)) = k$. Observe that $l(\sigma_{\vec{y}}, D) = l(\sigma_0, D)$. Moreover, from $x_{\mu, i_y} = 1$ it follows that $\mathcal{F}(l(\mu, D)) \neq k$. Since $\mathcal{F}(l(\mu, D)) = \mathcal{F}(l(\pi, D))$ this implies that π cannot be a tail-adjusted HUCF order for D. So, π can be chosen to be an HUCF order for D.

Define K_1, K_2, R_1, and M as follows

$$\begin{aligned} K_1 &= \{i \in D \big| \mathcal{F}(i) = k \text{ and } \mathcal{F}(j) = k \text{ for all } j \in D \text{ with } \sigma_0(j) \geq \sigma_0(i)\}, \\ K_2 &= \{i \in T_y \big| \mathcal{F}(i) = k \text{ and } \mathcal{F}(j) = k \text{ for all } j \in T_y \text{ with } \sigma_{\vec{y}}(j) \leq \sigma_{\vec{y}}(i)\}, \\ R_1 &= \{i \in D \big| \mathcal{F}(i) = k\} \supset K_1, \\ M &= \{i \in D \big| \mu(i) \geq \mu(l(\mu, R_1))\}. \end{aligned}$$

Note that $M \neq \emptyset$. Order μ' is obtained from μ by moving all jobs in K_2 to the tail of R_1. Figure 5.4 depicts the orders $\sigma_{\vec{y}}$, μ, and μ', and K_1, K_2, R_1, and M.

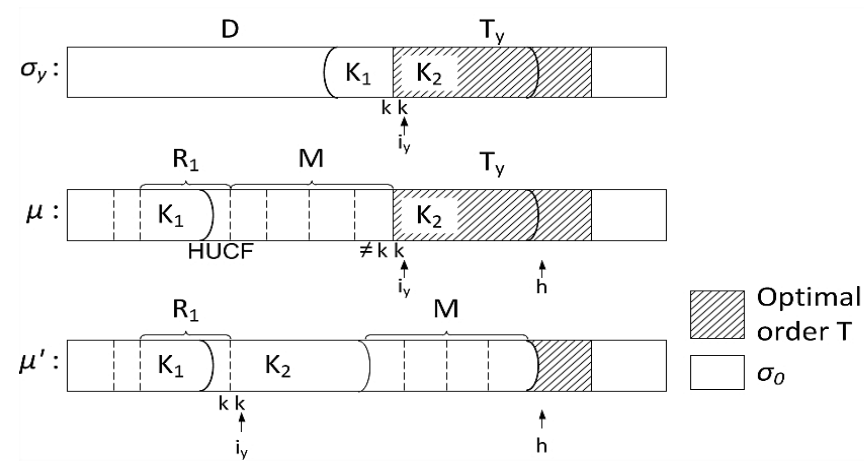

Figure 5.4: The orders $\sigma_{\vec{y}}$, μ and μ'.

To prove (5.11) we first prove that μ' is an admissible order for E and secondly, that

$$\sum_{j \in E} \alpha_{\mathcal{F}(j)} \left(C(\mu, j) - C(\mu', j)\right) \geq s_k \sum_{j \in T_y} \alpha_{\mathcal{F}(j)}. \tag{5.14}$$

Indeed, observe that if μ' is an admissible order for E, (5.14) implies (5.11) since

$$\begin{aligned} & v(E) - v(D) - \sum_{j \in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right) \\ \geq & \sum_{j \in E} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\mu', j)\right) - v(D) - \sum_{j \in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{j\in E} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\mu, j)\right) + \sum_{j\in E} \alpha_{\mathcal{F}(j)} \left(C(\mu, j) - C(\mu', j)\right) \\
&\quad - \sum_{j\in D} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\pi, j)\right) - \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right) \\
&\geq \sum_{j\in E} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\mu, j)\right) + s_k \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \\
&\quad - \sum_{j\in D} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\pi, j)\right) - \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right) \\
&= \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma, i) - C(\mu, j)\right) + s_k \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \\
&= \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_{\vec{y}}, i) - C(\mu, j)\right) + s_k \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \\
&\geq s_k(x_{\sigma_{\vec{y}}, i_y} - x_{\mu, i_y}) \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} + s_k \sum_{j\in T_y} \alpha_{\mathcal{F}(j)} \\
&= 0.
\end{aligned}
$$

Here the first inequality follows from the admissibility of μ' and the second from (5.14). The first equality follows from rewriting and the optimality of order π for D, the second from the fact that $C(\pi, i) = C(\mu, i)$ for all $i \in D$. The third equality holds by $C(\sigma, i) = C(\sigma_{\vec{y}}, i)$ for all $i \in T_y$. The last inequality holds due to (5.12). By assumption it holds that $x_{\sigma_{\vec{y}}, i_y} = 0$ and $x_{\mu, i_y} = 1$ such that the last equality holds.

First we prove that μ' is an admissible order for E. Clearly, $P(\mu', i) = P(\mu, i) = P(\sigma_0, i)$ for all $i \in N \setminus E$. So, it is sufficient to show that $C(\mu', i) \leq C(\sigma_0, i)$ for every $i \in N \setminus E$.

Let $h = f(\mu, S(\mu, K_2))$. Observe that h is not necessarily an element of T_y. For every $i \in S(\mu, K_2)$ it holds that

$$
\begin{aligned}
C(\mu, i) - C(\mu', i) &= \sum_{j\in \bar{P}(\mu, i)} (s_{\mathcal{F}(j)} x_{\mu, j} + p_{\mathcal{F}(j)}) - \sum_{j\in \bar{P}(\mu', i)} (s_{\mathcal{F}(j)} x_{\mu', j} + p_{\mathcal{F}(j)}) \\
&= \sum_{j\in \{i_y, h\}} (x_{\mu, j} - x_{\mu', j}) s_{\mathcal{F}(j)} \\
&= s_k + (x_{\mu, h} - x_{\mu', h}) s_{\mathcal{F}(h)}. \qquad (5.15)
\end{aligned}
$$

If $x_{\mu,h} = 1$, then for every $i \in S(\mu, K_2)$ it follows that

$$\begin{aligned} C(\mu', i) &= C(\mu, i) - s_k - (x_{\mu,h} - x_{\mu',h}) s_{\mathcal{F}(h)} \\ &\leq C(\mu, i) - s_k \\ &\leq C(\sigma_{\vec{y}}, i) \\ &\leq C(\sigma_0, i). \end{aligned}$$

Here the equality holds by (5.15). The first inequality holds by assumption, the second by (5.12), and the third by $\sigma_{\vec{y}} \in \mathcal{A}(E)$. Hence, μ' is an admissible order for E.

If $x_{\mu,h} = 0$ then by definition of K_2 it holds that $h \notin T_y$ and $T_y = K_2$. Hence, it follows that $C(\pi, i) = C(\mu, i)$ for all $i \in S(\sigma_0, E)$. Further it holds that $\mathcal{F}(h) = k$ and $x_{\mu',h} = 1$. Now, for all $i \in N \setminus E$,

$$C(\mu', i) = C(\mu, i) = C(\pi, i) \leq C(\sigma_0, i).$$

Here the first equality follows from (5.15), where $x_{\mu,h} = 0$, $x_{\mu',h} = 1$, and $\mathcal{F}(h) = k$. The inequality holds since $\pi \in \mathcal{A}(D)$. Hence, μ' is an admissible order for E.

It remains to prove that inequality (5.14) holds. Let γ be the time to process and setup all jobs in M when they are processed with respect to μ, i.e., $\gamma = \sum_{j \in M} (x_{\mu,j} s_{\mathcal{F}(j)} + p_{\mathcal{F}(j)})$. To show (5.14), we first show that

$$\gamma \alpha_k - p_k \sum_{j \in M} \alpha_{\mathcal{F}(j)} \geq 0. \tag{5.16}$$

Let π' be the order obtained from π by taking all jobs in R_1 behind M. Figure 5.5 depicts the two orders π and π'.

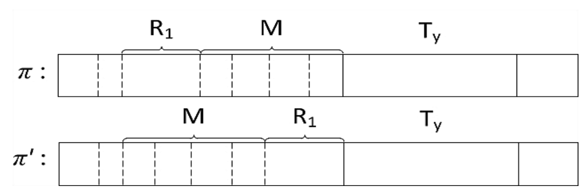

Figure 5.5: The orders π and π'.

Since π is optimal for D, π is not tail-adjusted HUCF, and since the number of setups in π is equal to the number of setups in π', order π' is admissible for D. Observe that for all $i \in D$

$$C(\pi', i) - C(\pi, i) = \begin{cases} 0 & \text{if } i \in D \backslash (M \cup R_1), \\ \gamma & \text{if } i \in R_1, \\ -(s_k + |R_1| p_k) & \text{if } i \in M. \end{cases} \tag{5.17}$$

Therefore,

$$\begin{aligned} & |R_1| \left(\gamma \alpha_k - p_k \sum_{j \in M} \alpha_{\mathcal{F}(j)} \right) - s_k \sum_{j \in M} \alpha_{\mathcal{F}(j)} \\ = \;& |R_1| \gamma \alpha_k - (s_k + |R_1| p_k) \sum_{j \in M} \alpha_{\mathcal{F}(j)} \\ = \;& \sum_{j \in R_1 \cup M} \alpha_{\mathcal{F}(j)} \left(C(\pi', j) - C(\pi, j) \right) \\ = \;& \sum_{j \in D} \alpha_{\mathcal{F}(j)} \left(C(\pi', j) - C(\pi, j) \right) \\ = \;& \sum_{j \in D} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\pi, j) \right) - \sum_{j \in D} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\pi', j) \right) \\ = \;& v(D) - \sum_{j \in D} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\pi', j) \right) \\ \geq \;& 0, \end{aligned}$$

where the second equality holds by (5.17) and the last equality follows from the fact that π is an optimal order for D. The inequality holds by the admissibility of π' for D. Hence,

$$\gamma \alpha_k - p_k \sum_{j \in M} \alpha_{\mathcal{F}(j)} \geq \frac{s_k \sum_{j \in M} \alpha_{\mathcal{F}(j)}}{|R_1|} \geq 0, \tag{5.18}$$

which proves (5.16).

With respect to (5.14), observe that for $i \in E$

$$C(\mu, i) - C(\mu', i) = \begin{cases} 0, & \text{if } i \in P(\mu, M), \\ -|K_2| p_k, & \text{if } i \in M, \\ \gamma + s_k, & \text{if } i \in K_2, \\ s_k + (x_{\mu,h} - x_{\mu',h}) s_j & \text{if } i \in T_y \setminus K_2. \end{cases} \tag{5.19}$$

Hence,

$$
\begin{aligned}
& \sum_{j \in E} \alpha_{\mathcal{F}(j)} \left(C(\mu, j) - C(\mu', j)\right) \\
= \;& \sum_{j \in M} \alpha_{\mathcal{F}(j)} \left(C(\mu, j) - C(\mu', j)\right) + \sum_{j \in K_2} \alpha_{\mathcal{F}(j)} \left(C(\mu, j) - C(\mu', j)\right) \\
& + \sum_{j \in T_y \setminus K_2} \alpha_{\mathcal{F}(j)} \left(C(\mu, j) - C(\mu', j)\right) \\
= \;& -|K_2| p_k \sum_{j \in M} \alpha_{\mathcal{F}(j)} + (\gamma + s_k) \sum_{j \in K_2} \alpha_{\mathcal{F}(j)} \\
& + (s_k + (x_{\mu,h} - x_{\mu',h}) s_h) \sum_{j \in T_y \setminus K_2} \alpha_{\mathcal{F}(j)} \\
\geq \;& -|K_2| p_k \sum_{j \in M} \alpha_{\mathcal{F}(j)} + (\gamma + s_k) \sum_{j \in K_2} \alpha_{\mathcal{F}(j)} + s_k \sum_{j \in T_y \setminus K_2} \alpha_{\mathcal{F}(j)} \\
= \;& |K_2| \left(\gamma \alpha_k - p_k \sum_{j \in M} \alpha_{\mathcal{F}(j)} \right) + s_k \sum_{j \in K_2} \alpha_{\mathcal{F}(j)} + s_k \sum_{j \in T_y \setminus K_2} \alpha_{\mathcal{F}(j)} \\
\geq \;& s_k \sum_{j \in T_y} \alpha_{\mathcal{F}(j)}.
\end{aligned}
$$

The second equality follows from (5.19). The first inequality follows from the fact that if $T_y \setminus K_2 \neq \emptyset$, then $x_{\mu,h} = 1$. The last inequality follows from (5.18). Hence, (5.14) is verified. This concludes the proof. □

Theorem 5.3.1 tells us that the set of stable and efficient allocations is nonempty. On the other hand, although $m^{\sigma_0}(v)$ is in the core, it is not very desirable for an effective sharing of the savings. For example, by this allocation rule the first job in the initial processing order obtains nothing although this may be a key job to obtain savings.

5.4 Ordered Family Sequencing Games

In this section we consider *ordered family sequencing situations*. A family sequencing situation is called *ordered* if the initial order σ_0 is family ordered. Note that since all family members are processed consecutively, the number of setups is minimized in σ_0.

Let $n = \sigma_0^{-1}(|N|)$. For a game $v \in TU^N$ the subgame $v^{-n} \in TU^{N\setminus\{n\}}$ is defined by

$$v^{-n}(S) = v(S),$$

for all $S \in 2^{N\setminus\{n\}}$. Hence, v^{-n} coincides with v for all coalitions where the last job (with respect to σ_0) does not participate. The next theorem shows, for ordered family sequencing situations, that v^{-n} is convex and that v has a component additive value structure.

Theorem 5.4.1. *Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be an ordered family sequencing situation with corresponding family sequencing game $v \in TU^N$. Let $n = \sigma_0^{-1}(|N|)$. Then,*

(i) the subgame $v^{-n} \in TU^{N\setminus\{n\}}$ is convex, and

(ii) $v(S) = \sum_{T \in S\setminus\sigma_0} v(T)$ for all $S \in 2^N$.

Proof. Set $f = |F|$. Let $F_k = \mathcal{F}^{-1}(k)$ be the set of family members of family $k \in F$ ordered in such a way that

$$\sigma_0(i) < \sigma_0(j)$$

whenever $i \in F_k, j \in F_l$ with $k, l \in F$ and $k < l$. Note that $n \in F_f$. Let $S \subset N$, ,$K^S \subset F$ denotes the set of families for which each member is in S, and $P^S \subset F$ denotes the set of families in S for which at least one family member is not in S. Clearly, a coalition S can be written as

$$S = \bigcup_{k \in K^S} F_k \cup \bigcup_{k \in P^S} G_k^S, \tag{5.20}$$

with $G_k^S \subsetneq F_k$ for all $k \in P^S$, where G_k^S is the set of members of family $k \in P^S$ which are in S.

Since the number of setups is minimized in σ_0, the number of setups equals f and

$$C(\sigma_0, n) = \min_{\sigma \in \Pi(N\setminus\{n\})} C(\sigma, n).$$

Consequently, each admissible order for an arbitrary coalition of $N \setminus \{n\}$ has to remain family ordered.

Consider the standard sequencing situation $(\hat{N}, \hat{\sigma}_0, \hat{p}, \hat{\alpha})$ with $\hat{N} = \{1, 2, \ldots, f-1\}$ and

$$\begin{aligned} \hat{\sigma}_0(k) &= k, \\ \hat{p}_k &= s_k + n_k p_k, \\ \hat{\alpha}_k &= n_k \alpha_k, \end{aligned}$$

for $k \in \hat{N}$. Denote the corresponding standard sequencing game by $\hat{v} \in TU^{\hat{N}}$. It follows from Curiel et al. (1989) that $\hat{v}$ is convex and $\hat{\sigma}_0$-component additive. Then explicitly using the fact that each admissible order for $N \backslash \{n\}$ is family ordered, one readily checks that for every $\hat{S} \subset \hat{N}$ it holds that

$$\hat{v}(\hat{S}) = v\left(\bigcup_{k \in \hat{S}} F_k\right). \tag{5.21}$$

(i) Let $i \in N \setminus \{n\}$ and consider $S \subset T \subset N \setminus \{n, i\}$. For convexity of v^{-n}, it suffices to prove that

$$v^{-n}(S \cup \{i\}) - v^{-n}(S) \leq v^{-n}(T \cup \{i\}) - v^{-n}(T),$$

or equivalently, that

$$v(S \cup \{i\}) - v(S) \leq v(T \cup \{i\}) - v(T). \tag{5.22}$$

From (5.20) and (5.21) and the fact that each admissible order for S has to remain family ordered it follows that

$$v(S) = v\left(\bigcup_{k \in K^S} F_k \cup \bigcup_{k \in P^S} G_k^S\right) = v\left(\bigcup_{k \in K^S} F_k\right) = \hat{v}(K^S).$$

The first equality follows from (5.20), the second from the fact that jobs in G^S are fixed, and the third from (5.21). Similarly, $v(T) = \hat{v}(K^T)$. Also note that $K^S \subset K^T$.

If $K^{S \cup \{i\}} = K^S$, then $v(S \cup \{i\}) = \hat{v}(K^S) = v(S)$ and (5.22) holds by monotonicity.

If $K^{S \cup \{i\}} \neq K^S$, then

$$S \cup \{i\} = \bigcup_{k \in K^S} F_k \cup F_{\mathcal{F}^{-1}(i)} \cup \bigcup_{k \in P^S \backslash \mathcal{F}^{-1}(i)} G_k^S,$$

and

$$T \cup \{i\} = \bigcup_{k \in K^T} F_k \cup F_{\mathcal{F}^{-1}(i)} \cup \bigcup_{k \in P^T \backslash \mathcal{F}^{-1}(i)} G_k^T.$$

Hence,

$$\begin{aligned} v(S \cup \{i\}) - v(S) &= \hat{v}(K^S \cup \mathcal{F}^{-1}(i)) - \hat{v}(K^S) \\ &\leq \hat{v}(K^T \cup \mathcal{F}^{-1}(i)) - \hat{v}(K^T) \\ &= v(T \cup \{i\}) - v(T). \end{aligned}$$

The inequality holds since $\hat{v}$ is convex.

(ii) Set $S \backslash \sigma_0 = \{S_1, \ldots, S_l\}$ such that $S_y \subset P(\sigma, S_{y+1})$ for every $y \in \{1, \ldots, l-1\}$. It follows that if $n \in S$, then n is in the last connected set, i.e. $n \in S_l$. Let, as before,

$$S_y = \bigcup_{k \in K_y^S} F_k \cup \bigcup_{k \in P_y^S} G_k^{S_y},$$

for all $y \in \{1, \ldots, l\}$. Note that $K^S \backslash \hat{\sigma}_0 = \cup_{y=1}^{l} K_y^S$ where K_y^S may be empty for some $y \in \{1, \ldots, l\}$.

First let $S \subset N \backslash \{n\}$. Since each admissible order remains family ordered, it suffices to prove *(ii)* for a coalition $S = \bigcup_{k \in K^S} F_k$ (i.e. with $P^S = \emptyset$). Then,

$$\begin{aligned} v(S) &= v\left(\bigcup_{k \in K^S} F_k\right) \\ &= \hat{v}(K^S) \\ &= \sum_{y=1}^{l} \hat{v}(K_y^S) \\ &= \sum_{y=1}^{l} v\left(\bigcup_{k \in K_y^S} F_k\right) \\ &= \sum_{y=1}^{l} v(S_y) \\ &= \sum_{T \in S \backslash \sigma_0} v(T). \end{aligned}$$

The third equality holds since $\hat{v}$ is $\hat{\sigma}_0$-component additive. This proves *(ii)* for $S \subset N \setminus \{n\}$.

Secondly, let $n \in S$. Let π be an optimal order for $\bigcup_{y=1}^{l-1} S_y$, μ an optimal order for S_l and define σ as follows

$$\sigma(i) = \begin{cases} \pi(i) & \text{if } i \in \bigcup_{y=1}^{l-1} S_y, \\ \mu(i) & \text{if } i \in S_l, \\ \sigma_0(i) & \text{otherwise.} \end{cases}$$

Using the fact that σ_0 is family ordered, one readily verifies that π is also family ordered and

$$C(\sigma, i) = C(\sigma_0, i) \tag{5.23}$$

for all $i \in N \setminus S$. Consequently, σ is also optimal for S. Thus,

$$\begin{aligned} v(S) &= \sum_{j \in S} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right) \\ &= \sum_{j \in \cup_{y=1}^{l-1} S_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right) + \sum_{j \in S_l} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\sigma, j)\right) \\ &= \sum_{j \in \cup_{y=1}^{l-1} S_y} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\pi, j)\right) + \sum_{j \in S_l} \alpha_{\mathcal{F}(j)} \left(C(\sigma_0, j) - C(\mu, j)\right) \\ &= v(\bigcup_{y=1}^{l-1} S_y) + v(S_l) \\ &= \sum_{y=1}^{l-1} v(S_y) + v(S_l) \\ &= \sum_{y=1}^{l} v(S_y) \\ &= \sum_{T \in S \setminus \sigma_0} v(T) \end{aligned}$$

where the fifth equality follows from (5.23). This finishes the proof of *(ii)*. □

In the following theorem, the relation between the core of an ordered family sequencing game v^n and the corresponding subgame v^{-n} is provided.

Theorem 5.4.2. *Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be an ordered family sequencing situation where $n = \sigma_0^{-1}(|N|)$, with corresponding family sequencing game $v \in TU^N$, and subgame $v^{-n} \in TU^{N \setminus n}$ with $x^{-n} \in Core(v^{-n})$. Then,*

$$\big(x^{-n}, v(N) - v(N \setminus \{n\})\big) \in Core(v).$$

Proof. It suffices to show that the marginal vector $m^{\sigma}(v)$, for an arbitrary $\sigma \in \Pi(N)$ such that $\sigma(n) = n$, belongs to the core of v. Let $\sigma \in \Pi(N)$. It suffices to prove that, for every $S \subset N$, it holds that

$$\sum_{j \in S} m_j^{\sigma}(v) \geq v(S).$$

Define $\sigma' \in \Pi(N \setminus \{n\})$ by $\sigma'(i) = \sigma(i)$ for all $i \in N \setminus \{n\}$.

Let $S \subset N$ be such that $n \notin S$. Then,

$$\sum_{j \in S} m_j^{\sigma}(v) = \sum_{j \in S} m_j^{\sigma'}(v^{-n}) \geq v^{-n}(S) = v(S),$$

where the inequality follows from Theorem 5.4.1*(i)*.

Let $S \subset N$ such that $n \in S$. Set $S \backslash \sigma_0 = \{S_1, \ldots, S_l\}$ such that $S_y \subset P(\sigma, f(\sigma, S_{y+1}))$ for every $y \in \{1, \ldots, l-1\}$. It follows that $n \in S_l$. Choose $x \in Core(v)$. Then,

$$\begin{aligned}
\sum_{j \in S} m_j^{\sigma}(v) &= \sum_{y=1}^{l-1} \sum_{j \in S_y} m_j^{\sigma}(v) + \sum_{j \in S_l} m_j^{\sigma}(v) \\
&\geq \sum_{y=1}^{l-1} v(S_y) + \sum_{j \in S_l} m_j^{\sigma}(v) \\
&= \sum_{y=1}^{l-1} v(S_y) + v(N) - v(N \setminus S_l) \\
&\geq \sum_{y=1}^{l-1} v(S_y) + \sum_{j \in N} x_j - \sum_{j \in N \setminus S_l} x_j \\
&= \sum_{y=1}^{l-1} v(S_y) + \sum_{j \in S_l} x_j \\
&\geq \sum_{y=1}^{l-1} v(S_y) + v(S_l) \\
&= v(S).
\end{aligned}$$

The first inequality follows from Theorem 5.4.1*(i)*. The second and third inequalities are due to the fact that $x \in Core(v)$. The last equality follows from Theorem 5.4.1*(ii)*. □

Theorem 5.4.1 and Theorem 5.4.2 allows us to provide a suitable solution concept for family ordered sequencing situations. For this purpose we use the Shapley value (Shapley (1953)). The Shapley value is defined as the average of all marginal vectors, i.e.,

$$\Phi(v) = \frac{1}{|N|!} \sum_{\sigma \in \Pi(N)} m^{\sigma}(v),$$

for all $v \in TU^N$. In Corollary 5.4.3 a Shapley-based core allocation is constructed for ordered family sequencing games.

Corollary 5.4.3. *Let $\Sigma(N) = (N, F, \mathcal{F}, \sigma_0, s, p, \alpha)$ be an ordered family sequencing situation with corresponding family sequencing game $v \in TU^N$. Let $n = \sigma_0^{-1}(|N|)$. Then,*

$$\big(\Phi(v^{-n}), v(N) - v(N \setminus \{n\})\big) \in Core(v).$$

The following example illustrates that, in general, the solution concept from Corollary 5.4.3 is not efficient for family sequencing games.

Example 5.4.1. Reconsider the family sequencing game of Example 5.3.1 with set of jobs $N = \{1, 2, 3, 4, 5\}$ and $v(N) = 72$. Note that σ_0 is not family ordered. Then,

$$\big(\Phi(v^{-n}), v(N) - v(N \setminus \{n\})\big) = \big(6\tfrac{3}{4}, 18\tfrac{3}{4}, 18\tfrac{3}{4}, 14\tfrac{3}{4}, 13\big) \notin Core(v).$$

This can be seen from the fact $v(N) = v(\{2345\})$ such that $x_1 = 0$ for all $x \in Core(v)$. ♢

Bibliography

Ahn, B. and J. Hyun (1990). Single facility multi-class job scheduling. *Computers and Operations Research*, **17**, 265–272.

Allahverdi, A., J. N. Gupta, and T. Aldowaisan (1999). A review of scheduling research involving setup considerations. *OMEGA, The International Journal of Management Science*, **27**, 219–239.

Allahverdi, A., C. Ng, T. Cheng, and M. Kovalyov (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, **187**, 985–1032.

Ambec, S. and A. Dinar (2009). Hot stuff: Would climate change alter transboundary water sharing agreements. In: *ISA's 50th Annual Convention New York City, NY, February*, pp. 15–18.

Ambec, S. and L. Ehlers (2008). Sharing a river among satiable agents. *Games and Economic Behavior*, **64**, 35 – 50.

Ambec, S. and Y. Sprumont (2002). Sharing a River. *Journal of Economic Theory*, **107**, 453 – 462.

Aumann, R. and M. Maschler (1985). Game theoretic analysis of a bankruptcy problem from the Talmud. *Journal of Economic Theory*, **36**, 195–213.

Beal, S., A. Ghintran, E. Remila, and P. Solal (2013). The river sharing problem: A survey. *International Game Theory Review*, **15**.

Bergantiños, G., L. Lorenzo, and S. Lorenzo-Freire (2010). A characterization of the proportional rule in multi-issue allocation situations. *Operations Research Letters*, **38**, 17 – 19.

Borm, P., M. Fiestras-Janeiro, H. Hamers, E. Sánchez, and M. Voorneveld (2002). On the convexity of games corresponding to sequencing situations with due dates. *European Journal of Operational Research*, **136**, 616–634.

Borm, P., H. Hamers, and R. Hendrickx (2001). Operations research games: a survey. *TOP*, **9**, 139–216.

Brink, R. van den, A. Estévez Fernández, G. van der Laan, and N. Moes (2014). Independence Axioms for Water Allocation. *Social Choice and Welfare*, **43**, 173–194.

Brink, R. Van den, G. Van der Laan, and N. Moes (2012). Fair agreements for sharing international rivers with multiple springs and externalities. *Journal of Environmental Economics and Management*, **63**, 388–403.

Brink, R. van den, G. van der Laan, and V. Vasil'ev (2003). Harsanyi solutions in line-graph games. Technical Report 03-076/1, Tinbergen Institute Discussion Paper.

Brink, R. van den, G. van der Laan, and V. Vasil'ev (2007). Component efficient solutions in line-graph games with applications. *Economic Theory*, **33**, 349–364.

Bruno, G., A. Genovese, and G. Improta (2011). Routing problems: a historical perspective. *BSHM Bulletin: Journal of the British Society for the History of Mathematics*, **26**, 118–127.

Bruno, J. and R. Sethi (1978). Task sequencing in a batch environment with setup times. *Foundations of Control Engineering*, **3**, 105–117.

Calleja, P., P. Borm, and R. Hendrickx (2005). Multi-issue allocation situations. *European Journal of Operational Research*, **164**, 730–747.

Carraro, C., C. Marchiori, and A. Sgobbi (2005). Applications of negotiation theory to water issues. FEEM Working Papers 2005.65, Fondazione Eni Enrico Mattei, Italy.

Condorelli, D. (2013). Market and non-market mechanisms for the optimal allocation of scarce resources. *Games and Economic Behavior*, **82**, 582–591.

Cruijssen, F. (2006). *Horizontal cooperation in transport and logistics*. Ph. D. thesis, Tilburg University, Tilburg, The Netherlands.

Curiel, I. (2008). Cooperative Combinatorial Games. In: *Pareto Optimality, Game Theory And Equilibria*, Volume 17 of *Springer Optimization and Its Applications*, pp. 131–157. Springer New York.

Curiel, I., H. Hamers, and F. Klijn (2002). Sequencing games: a survey. In: *Chapters in Game Theory*, pp. 27–50. Springer.

Curiel, I., M. Maschler, and S. Tijs (1987). Bankruptcy games. *Zeitschrift für Operations Research*, **31**, 143–159.

Curiel, I., G. Pederzoli, and S. Tijs (1989). Sequencing games. *European Journal of Operational Research*, **40**, 344–351.

Dinar, A. (2004). Exploring transboundary water conflict and cooperation. *Water Resources Research*, **40**, W05S01.

Dinar, A. and G. S. Nigatu (2013). Distributional considerations of international water resources under externality: The case of Ethiopia, Sudan and Egypt on the Blue Nile. *Water Resources and Economics*, **2**, 1–16.

Dinar, S. (2007). *International water treaties: Negotiation and cooperation along transboundary rivers*. Routledge Studies in the Modern World Economy. Taylor & Francis.

Dobson, G., S. Karmarkar, and J. L. Rummel (1987). Batching to minimize flow times on on one machine. *Management Science*, **33**, 784–799.

Estévez Fernández, A. (2008). *A Game Theoretical Approach to Sharing Penalties and Rewards in Projects*. Ph. D. thesis, Tilburg University, Tilburg, The Netherlands.

Fiestras-Janeiro, M., I. Garca-Jurado, A. Meca, and M. Mosquera (2011). Cooperative game theory and inventory management. *European Journal of Operational Research*, **210**, 459 – 466.

Gillies, D. B. (1953). *Some theorems on n-person games*. Ph. D. thesis, Princeton University Press, Princeton, New Jersey.

Gordon, V. and V. Strusevich (2009). Single machine scheduling and due date assignment with positionally dependent processing times. *European Journal of Operational Research*, **198**, 57–62.

Granot, D. and G. Huberman (1982). The relationship between convex games and minimal cost spanning tree games: A case for permutationally convex games. *SIAM Journal of Algorithms and Discrete Methods*, **3**, 288–292.

Groote Schaarsberg, M. (2014). *Interactive Operational Decision Making*. Ph. D. thesis, Tilburg University, Tilburg, The Netherlands.

Grundel, S., P. Borm, and H. Hamers (2013a). Resource allocation games: a compromise stable extension of bankruptcy games. *Mathematical Methods of Operations Research*, **78**, 149–169.

Grundel, S., P. Borm, and H. Hamers (2013b). Resource allocation games with concave reward functions. CentER Discussion Paper 2013-070, Tilburg University, Tilburg, The Netherlands.

Grundel, S., B. Çiftçi, P. Borm, and H. Hamers (2013). Family sequencing and cooperation. *European Journal of Operational Research*, **226**, 414–424.

Hamers, H., P. Borm, and S. Tijs (1995). On games corresponding to sequencing situations with ready times. *Mathematical Programming*, **70**, 1–13.

Harris, M. and R. M. Townsend (1981). Resource allocation under asymmetric information. *Econometrica*, **49**, 33–64.

Herings, P., G. van der Laan, and D. Talman (2008). The average tree solution for cycle-free graph games. *Games and Economic Behavior*, **62**, 77–92.

Jägerskog, A. (2007). Why states co-operate over shared water: the water negotiations in the Jordan River basin. In: *Water Resources in the Middle East*, Volume 2, pp. 195–202. Springer Berlin Heidelberg.

Kaminski, M. (2000). Hydraulic rationing. *Mathematical Social Sciences*, **40**, 131–155.

Karsten, F. (2013). *Resource pooling games*. Ph. D. thesis, Technische Universiteit Eindhoven.

Khmelnitskaya, A. B. (2010). Values for rooted-tree and sink-tree digraph games and sharing a river. *Theory and Decision*, **69**, 657–669.

Kilgour, D. M. and A. Dinar (2001). Flexible water sharing within an international river basin. *Environmental and Resource Economics*, **18**, 43–60.

Klijn, F. and E. Sánchez (2006). Sequencing games without initial order. *Mathematical Methods of Operations Research*, **63**, 53–62.

Koster, M. (1999). *Cost Sharing in Production Situations and Network Exploitation*. Ph. D. thesis, Tilburg University, Tilburg, The Netherlands.

Koulamas, C., S. Gupta, and G. J. Kyparisis (2010). A unified analysis for the single-machine scheduling problem with controllable and non-controllable variable job processing times. *European Journal of Operational Research*, **205**, 479–482.

Kuhn, H. W. and A. W. Tucker (1951). Nonlinear Programming. In: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, California, pp. 481–492. University of California Press.

Langar, R., S. Secci, R. Boutaba, and G. Pujolle (2015). An operations research game approach for resource and power allocation in cooperative Femtocell networks. *IEEE Transactions on Mobile Computing*, **14**, 675–687.

Le Breton, M., G. Owen, and S. Weber (1992). Strongly balanced cooperative games. *International Journal of Game Theory*, **20**, 419–427.

Lohmann, E., P. Borm, and M. Slikker (2014). Sequencing situations with Just-in-Time arrival and related games. *Mathematical Methods of Operations Research*, **80**, 285–305.

Madani, K. (2010). Game theory and water resources. *Journal of Hydrology*, **381**, 225–238.

Mason, A. and E. Anderson (1991). Minimizing flow time on a single machine with job classes and setup times. *Naval Research Logistics*, **38**, 333–350.

Moes, N. (2013). *Cooperative Decision Making in River Water Allocation Problems*. Ph. D. thesis, Vrije Universiteit, Amsterdam, The Netherlands.

Monma, C. and C. Potts (1989). On the complexity of scheduling with batch setup times. *Operations Research*, **37**, 798–804.

Mosheiov, G. and D. Oron (2008). A single machine batch scheduling problem with bounded batch size. *European Journal of Operational Research*, **187**, 1069–1079.

Moulin, H. (1991). *Axioms of cooperative decision making*. Cambridge University Press.

Myerson, R. B. (1979). Incentive compatibility and the bargaining problem. *Econometrica*, **47**, 61–73.

Ocetkiewicz, K. (2010). A FPTAS for minimizing total completion time in a single machine time-dependent scheduling problem. *European Journal of Operational Research*, **203**, 316–320.

O'Neill, B. (1982). A problem of rights arbitration from the Talmud. *Mathematical Social Sciences*, **2**, 345–371.

Ostrom, E., R. Gardner, J. Walker, and A. Agrawal (1994). *Rules, games, and common-pool resources*. University of Michigan Press.

Owen, G. (1975). On the core of linear production games. *Mathematical programming*, **9**, 358–370.

Pálvölgyi, D., H. J. M. Peters, and A. J. Vermeulen (2010). A strategic approach to estate division problems with non-homogenous preferences. METEOR 10/036, Maastricht University School of Business and Economics, Maastricht, The Netherlands.

Parrachino, I., A. Dinar, and F. Patrone (2006). Cooperative game theory and its application to natural, environmental, and water resource issues: 3. application to water resources. Policy Research Working Paper Series 4074, The World Bank.

Potters, J., I. Curiel, and S. Tijs (1992). Traveling salesman games. *Mathematical Programming*, **53**, 199–211.

Potters, J. and S. Tijs (1994). On the locus of the nucleolus. In: N. Megiddo (Ed.), *Essays in Game Theory: in honor of Michael Maschler*, Berlin, pp. 193–203. Springer-Verlag.

Potts, C. and M. Kovalyov (2000). Scheduling with batching: A review. *European Journal of Operational Research*, **120**, 228–249.

Potts, C. and L. Van Wassenhove (1992). Integrating scheduling with batching and lot-sizing: a review of algorithms and complexity. *Journal of the Operational Research Society*, **43**, 395–406.

Quant, M., P. Borm, H. Reijnierse, and S. van Velzen (2005). The core cover in relation to the nucleolus and the Weber set. *International Journal of Game Theory*, **33**, 491–503.

Ransmeier, J. (1942). *The Tennessee Valley Authority: A case study in the economics of multiple purpose stream planning*. Vanderbilt University Press.

Read, L., K. Madani, and B. Inanloo (2014). Optimality versus stability in water resource allocation. *Journal of environmental management*, **133**, 343–354.

Salman, S. M. (2007). The Helsinki rules, the UN Watercourses Convention and the Berlin rules: perspectives on international water law. *Water Resources Development*, **23**, 625–640.

Santos, C. and M. Magazine (1985). Batching in single operation manufacturing systems. *Operations Research Letters*, **4**, 99–103.

Schmeidler, D. (1969). The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, **17**, 1163–1170.

Shapley, L. (1953). A value for n-person games. *Annals of Mathematics Studies*, **28**, 307–317.

Straffin, P. and J. Heaney (1981). Game theory and the Tennessee Valley authority. *International Journal of Game Theory*, **10**, 35–43.

Thomson, W. (2003). Axiomatic and game theoretic analysis of bankruptcy and taxation problems: a survey. *Mathematical Social Sciences*, **45**, 249–297.

Thomson, W. (2015). Axiomatic and game-theoretic analysis of bankruptcy and taxation problems: An update. *Mathematical Social Sciences*, **74**, 41–59.

Tijs, S. (1981). Bounds for the core and the τ-value. In: O. Moeschlin and D. Pallaschke (Eds.), *Game Theory and Mathematical Economics*. North-Holland, Amsterdam, 123-132.

Tijs, S. and F. Lipperts (1982). The hypercube and the core cover of n-person cooperative games. *Cahiers du Centre d'Études de Recherche Opérationelle*, **24**, 27–37.

Velzen, B. van (2006). Sequencing games with controllable processing times. *European Journal of Operational Research*, **172**, 64–85.

Wan, G. and B. Yen (2009). Single machine scheduling to minimize total weighted earliness subject to minimal number of tardy jobs. *European Journal of Operational Research*, **195**, 89–97.

Wang, Y. (2011). Trading water along a river. *Mathematical Social Sciences*, **61**, 124–130.

Weikard, H.-P., E. Ansink, and M. Gengenbach (2013). River sharing and water trade. Annual Conference 2013 (Duesseldorf): Competition Policy and Regulation in a Global Economic Order 79931, Verein fr Socialpolitik/ German Economic Association.

Young, H. (1988). Distributive justice of taxation. *Journal of Economic Theory*, **44**, 321–335.

Young, H. (1995). *Equity, in theory and practice*. Princeton: Princeton University Press.

Zara, S., A. Dinar, and F. Patrone (2006). Cooperative game theory and its application to natural, environmental, and water resource issues: 2. application to natural and environmental resources. Policy Research Working Paper Series 4073, The World Bank.

Author index

Agrawal, A., 36, 55
Ahn, B., 89
Aldowaisan, T., 89
Allahverdi, A., 89
Ambec, S., 7, 36, 55–59
Anderson, E., 89
Ansink, E., 36
Aumann, R., 11, 13, 32, 50

Beal, S., 57
Bergantiños, G., 12
Borm, P., 2, 9, 12, 13, 25, 28, 33, 35, 90, 91
Boutaba, R., 7
Brink, R., van den, 36, 57
Bruno, G., 2
Bruno, J., 89

Calleja, P., 12
Carraro, C., 13, 36
Cheng, T., 89
Ciftçi, B., 9, 90
Condorelli, D., 36
Cruijssen, F., 1
Curiel, I., 2–4, 8, 52, 91, 116

Dinar, A., 7, 13, 36, 56, 57
Dinar, S., 36, 55
Dobson, G., 93

Ehlers, L., 36, 57, 59
Estévez Fernández, A., 6, 57

Fiestras-Janeiro, M., 2, 91

Garca-Jurado, I., 2
Gardner, R., 36, 55
Genovese, A., 2
Ghintran, A., 57
Gillies, D., 2
Gordon, V., 90
Granot, D., 2, 91, 105
Groote Schaarsberg, M., 1
Grundel, S., 9, 12, 35, 90
Gupta, J., 89
Gupta, S., 90

Hamers, H., 2, 9, 12, 13, 35, 90, 91
Harris, M., 36
Heaney, J., 13
Hendrickx, R., 2, 12, 13
Herings, P., 57
Huberman, G., 2, 91, 105
Hyun, J., 89

Improta, G., 2
Inanloo, B., 7

Jägerskog, A., 56

Kaminski, M., 12
Karmarkar, S., 93
Karsten, F., 1
Khmelnitskaya, A., 57
Kilgour, D., 7, 57
Klijn, F., 2, 91
Koster, M., 2
Koulamas, C., 90

Kovalyov, M., 89
Kuhn, H., 39, 61
Kyparisis, G., 90

Laan, G., van der, 57
Langar, R., 7
Le Breton, M., 104
Lipperts, F., 25
Lohmann, E., 91
Lorenzo, L., 12
Lorenzo-Freire, S., 12

Madani, K., 7, 13
Magazine, M., 90, 91, 93
Marchiori, C., 13, 36
Maschler, M., 11, 13, 32, 50, 52
Mason, A., 89
Meca, A., 2
Moes, N., 7, 56, 57
Monma, G., 89
Mosheiov, G., 89
Mosquera, M., 2
Moulin, H., 11
Myerson, R., 36

Ng, C., 89
Nigatu, G., 56

O'Neill, B., 4, 9, 11, 36
Ocetkiewicz, K., 90
Oron, D., 89
Ostrom, E., 36, 55
Owen, G., 2, 104

Parrachino, I., 13, 36, 57
Patrone, F., 13, 36, 57
Pederzoli, G., 3, 4, 8, 91, 116
Peters, H., 36
Potters, J., 2, 34, 52
Potts, C., 89, 90
Pujolle, G., 7
Pálvölgyi, 36

Quant, M., 25, 28, 33

Ransmeier, J., 13
Read, L., 7
Reijnierse, H., 25, 28, 33
Remila, E., 57
Rummel, J., 93

Salman, S., 56
Santos, C., 90, 91, 93
Schmeidler, D., 2, 13, 32, 37, 50
Secci, S., 7
Sethi, R., 89
Sgobbi, A., 13, 36
Shapley, L., 2, 91, 120
Slikkers, M., 91
Solal, P., 57
Sprumont, Y., 7, 36, 55–58
Straffin, P., 13
Strusevich, V., 90
Sánchez, E., 91

Talman, D., 57
Thomson, W., 5, 11
Tijs, S., 2–4, 8, 25, 34, 52, 91, 116
Townsend, R., 36
Tucker, A., 39, 61

Vasil'ev, V., 57

Velzen, B. van, 25, 28, 33, 91, 106
Vermeulen, A., 36
Voorneveld, M., 91

Walker, J., 36, 55
Wan, G., 89
Wang, Y., 36, 57
Wassenhove, L. van, 90
Weber, S., 104
Weikard, H., 36

Yen, B., 89
Young, H., 11, 12, 35

Zara, S., 13